An agent-based approach to customer crowd-shipping

by

Phillip Christian Malan

Thesis presented in partial fulfilment of the requirements for the degree of
Master of Engineering (Industrial Engineering)
in the Faculty of Engineering at Stellenbosch University

Supervisor: Dr Christa Searle
Co-supervisor: Prof JH van Vuuren

March 2022

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: November 18, 2021

Abstract

The challenge of effective last-mile deliveries is progressively becoming more important with the acceleration in the e-commerce industry that is accompanied by a growing number of doorstep deliveries. Crowd logistics provides innovative solutions whereby ordinary people become involved in the execution of logistics operations. A particular crowd logistics initiative, referred to as *customer crowd-shipping*, recently gained interest from researchers after initial implementations thereof had been performed by companies such as Walmart and Amazon. The approach involves the use of a retailer's in-store customers, in addition to regular delivery vehicles, for delivering orders to online customers. Such in-store customers, referred to as *occasional drivers*, are offered incentives to deliver orders on their way home after visiting the retailer.

In this thesis, an agent-based simulation model is proposed for studying the highly dynamic working of the customer crowd-shipping initiative. The model encompasses a traditional last-mile delivery system, complemented by the ability to utilise autonomous occasional drivers. The modelled traditional last-mile delivery system consists of a dedicated fleet of delivery vehicles serving online customers from a single depot. The execution of deliveries is formulated as a vehicle routing problem and subsequently solved by means of well-known vehicle routing heuristics. In addition, the occasional drivers are modelled as autonomous agents who have the ability to act outside of the control of the retailer. Rather than being assigned to particular orders, occasional drivers are presented with potential orders from which they may select an order suitable for them to deliver. Their decision to participate is modelled based on self-interest, where an occasional driver agent aims to maximise the difference between the incentive offered and his or her perceived value of the additional time required to deliver the order.

An integrated approach to customer crowd-shipping is developed in order to consider the benefits for both the retailer and occasional drivers. This includes an incentive scheme and a method for identifying online customers as candidates for crowd-shipping. The latter involves the dynamic calculation of the company's cost to serve an individual customer, which is determined for all online customers. Finally, user-friendly access to the agent-based simulation model is facilitated by a graphical user interface.

The proposed model is subjected to systematic verification, ensuring the correct functioning and integration of its subcomponents. Moreover, the model is evaluated under various operating conditions to gain a deeper understanding of the crowd-shipping initiative, while simultaneously validating the model as adequate. In particular, parameter variation, sensitivity analyses, and scenario analyses are conducted, followed by face validation by subject matter experts.

The results of the various analyses indicate that customer crowd-shipping may successfully function as an extension to an existing last-mile delivery system, with the potential of reducing both the total delivery cost and customer waiting time. These benefits are, however, shown to be influenced by the incentive scheme and the strategy by which online customers are selected as crowd-shipping candidates. Finally, it is deduced that the maturity of the customer crowd-shipping system and the occasional population's perceived value of time influence the performance of the customer crowd-shipping model.

Opsomming

Die uitdaging van doeltreffende laaste-myl aflewerings word geleidelik belangriker met die versnelling in die e-handelsbedryf wat gepaard gaan met 'n groeiende aantal voorstoepaflewerings. Skare-logistiek bied innoverende oplossings waardeur gewone mense betrokke raak by die uitvoering van logistieke bedrywighede. 'n Sekere skare-logistieke inisiatief, waarna verwys word as *kliënte-skareversending*, het onlangs belangstelling by navorsers ontlok nadat aanvanklike implementering daarvan deur maatskappye soos Walmart en Amazon plaasgevind het. Die benadering behels die gebruik van 'n kleinhandelaar se in-winkel kliënte, benewens normale afleweringsvoertuie, om bestellings by aanlynkliënte af te lewer. Sulke in-winkel kliënte, na wie daar ook verwys word as *geleentheidsbestuurders*, word aansporings gebied om bestellings op pad huis toe af te lewer nadat hulle die kleinhandelaar besoek het.

In hierdie tesis word 'n agent-gebaseerde simulasiemodel voorgestel vir die bestudering van die hoogs-dinamiese werking van die kliënte-skareversendingsinisiatief. Die model sluit 'n tradisionele laaste-myl afleweringstelsel in, aangevul deur die mootlikheid om outonome geleentheidsbestuurders te gebruik. Die gemodelleerde tradisionele laaste-myl afleweringstelsel bestaan uit 'n toegewyde vloot afleweringsvoertuie wat aanlynkliënte vanaf 'n enkele depot bedien. Die uitvoering van aflewerings word as 'n voertuig-roeteringsprobleem geformuleer en vervolgens deur middel van bekende voertuig-roeteringsheuristieke opgelos. Daarbenewens word die geleentheidsbestuurders as outonome agente gemodelleer wat oor die vermoë beskik om buite die beheer van die kleinhandelaar op te tree. Eerder as om aan spesifieke bestellings toegewys te word, word geleentheidsbestuurders potensiële bestellings aangebied waaruit hulle een kan kies wat geskik is om deur hulle afgelewer te word. Hul besluit om deel te neem berus op eiebelang, waar 'n geleentheidsbestuurder-agent poog om die verskil tussen die aansporing wat aangebied word en sy of haar waargenome waarde van die bykomende tyd wat benodig word om die bestelling af te lewer, te maksimeer.

'n Geïntegreerde benadering tot kliënte-skareversending word ontwikkel om die voordele vir beide die kleinhandelaar en geleentheidsbestuurders te oorweeg. Dit sluit 'n aansporingskema in sowel as 'n metode om aanlynkliënte as kandidate vir skareversending te identifiseer. Laasgenoemde behels die dinamiese berekening van die maatskappy se koste om 'n individuele kliënt te bedien, wat vir alle aanlynkliënte bepaal word. Laastens word gebruikersvriendelike toegang tot die agent-gebaseerde simulasiemodel deur 'n grafiese gebruikerskoppelvlak moontlik gemaak.

Die voorgestelde model word aan sistematiese verifikasie onderwerp, wat die korrekte funksionering en integrasie van die deelkomponente daarvan verseker. Boonop word die model onder verskeie bedryfstoestande geëvalueer om 'n dieper begrip van die kliënte-skareversendingsinisiatief te verkry, terwyl die model terselfdertyd as voldoende bekragtig word. In die besonder word parametervariasie, sensitiwiteitsanalises en scenario-ontledings uitgevoer, gevolg deur sigvalidering deur vakkundiges.

Die resultate van die verskillende ontledings dui daarop dat kliënte-skareversending suksesvol as 'n uitbreiding van 'n bestaande laaste-myl afleweringstelsel kan funksioneer, met die potensiaal om beide die totale afleweringskoste en kliëntewagtyd te verminder. Daar word egter getoon dat hierdie voordele beïnvloed word deur die aansporingskema en die strategie waardeur aanlyn-kliënte as skareversendingkandidate gekies word. Laastens word afgelei dat die volwassenheid van die kliënt-skareversendingstelsel en die bevolking geleentheidsbestuurders se waargenome waarde van tyd die prestasie van die kliënte-skareversendingsmodel beïnvloed.

Acknowledgements

The author wishes to acknowledge the following people and institutions for their various contributions towards the completion of this work:

- Dr Christa Searle, for her excellent supervision over the course of the past two years. Although the post-graduate journey was by no means smooth and circumstances were often challenging, you were unwavering in your commitment to guide my efforts in the right direction. Thank you for the large degree of trust that you showed throughout the process. Freedom was afforded to me both within the context of my research and the post-graduate experience as a whole. Finally, the braais and conversations with you and Kit will be remembered fondly, with the tea-time treats by Kit always proving to be a highlight.

- Prof Jan van Vuuren, for his excellent guidance as co-supervisor and mentor. Your willingness to step in and play an advisory role is greatly appreciated. We are all indebted to your excellent leadership in cultivating a wonderful learning environment — by not only teaching us everything you know, but also allowing us to share our knowledge.

- The members of *Stellenbosch Unit of Operations Research in Engineering* (SUnORE), for cultivating the ideal environment for conducting high-quality research while maintaining a social post-graduate life. The broad range of skills and the mindset of excellence will be carried with me throughout my future endeavours. I will always hold our tea-time conversations dear, with topics ranging from philosophical to political to ridiculous. You helped me to find my voice and made me feel like I belong. I will forever be grateful.

- Daan and Tharin Malan, for your support and love throughout my studies and my life. A special thanks to my mom, for the years invested into gifting us a high-quality education with a focus on self-driven exploration and a love for mathematics — I cherish every moment and will forever be grateful. A special thanks to my dad for being the most consistent and reliable person I know and for having provided the structure required for me to pursue my goals. Thank you both for all the sacrifices made for your family's sake — no words can adequately express my gratitude.

- Tannie Karina, for being the backbone of the Industrial Engineering department. The quick chats at the reception desk served as the ideal resting point for the daily journey to the fourth floor. Your infectious, light-hearted spirit and kind words were sources of encouragement throughout my time as an Industrial Engineering student.

- Lisa, Lizanne, and Lianri Malan, for being incredible siblings, constantly inspiring me grow. I look up to each one of you.

- The Rooted Team, Neil and Chrisjan, for making the past two years an incredible journey. Thank you for contributing to my decision to pursue post-graduate studies, for supporting me throughout the process, and for making it two of the best years of my life. My post-graduate journey would certainly not have been the same without your influence.

- Vera-Lynne Smit, for being the most amazing companion during the course of my masters. Thank you for your ever-eager ear in listening to my thoughts, for every moment spent reading my thesis, and for the immensely valuable inputs you contributed to my research. Thank you for standing by my side throughout a challenging year — your belief, encouragement, and love truly means the world to me. I am blessed to have you in my life.

The Industrial Engineering Department and, in particular, SUnORE, is thanked for the use of their office space. Finally, the Stellenbosch University Postgraduate Office and SUnORE are respectively thanked for their financial support in the form of post-graduate bursaries.

Table of Contents

List of Acronyms

ABM: Agent-based modelling

ANOVA: Analysis of variance

B2B: Business-to-business

B2C: Business-to-consumer

CVRP: Capacitated vehicle routing problem

C2C: Consumer-to-consumer

DC: Distribution centre

DCVRP: Distance-constrained vehicle routing problem

GAP: General assignment problem

GDP: Gross domestic product

GUI: Graphical user interface

ICT: Information and communications technology

KPI: Key performance indicator

MRS: Marginal rate of substitution

OD: Occasional driver

TSP: Travelling salesman problem

VRP: Vehicle routing problem

VRPB: Vehicle routing problem with backhauls

VRPB: Vehicle routing problem with backhauls with time windows

VRPOD: Vehicle routing problem with occasional drivers

VRPPD: Vehicle routing problem with pickup and delivery

VRPPDTW: Vehicle routing problem with pickup and delivery with time windows

VRPTW: Vehicle routing problem with time windows

WTP: Willingness to pay

WTA: Willingness to accept

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

Contents

1.1 Background

The Internet has become a fundamental and irreplaceable element in society. The increased convenience and accessibility brought about by high-speed mobile internet connection have opened a world of opportunities to consumers and businesses alike. The Internet provides a platform where people socialise, share about their lives, stay up to date with news, and do their shopping. This has granted businesses the opportunity to reach millions of people on a single website. In 2020, the electronic commerce (e-commerce) industry generated sales of $4.2 trillion globally and it is predicted that roughly 21% of all retail sales will occur through online platforms by 2024 [51, 52]. The true growth of the e-commerce industry may be significantly greater, however, as a result of the COVID-19 pandemic. Due to many global social distancing measures and lock-downs, people have been restricted from visiting various brick-and-mortar stores and are urged to rather shop online. Even though consumer spending fell by 7.5% in March 2020 from that of a year earlier, the revenue gained by Amazon through online sales increased by 26% during the same time period [137]. All the aforementioned factors have contributed to a major shift in business, with various brick-and-mortar retailers diversifying their sales channels through the addition of an online shopping platform.

The growth of the e-commerce industry is naturally succeeded by an increased number of parcel deliveries. This poses major operational challenges for both purely e-commerce and omni-channel businesses, relating to the planning and execution of deliveries to their customers. In particular, for businesses that target end-consumers through an online channel, the planning, execution, and control of doorstep deliveries are critical aspects of the supply chain. This terminal link in the supply chain is referred to as the *last mile*, and is considered to be one of the most challenging facets of a supply chain [77]. The efficient and convenient execution of such home deliveries, referred to as *last-mile deliveries*, require a high level of coordination and are often extremely expensive [151]. In fact, it is estimated that up to 30–70% of a supply chain's total logistics costs are attributed to last-mile deliveries [19]. Moreover, the problem of expensive last-mile deliveries is further exacerbated by consumers' extreme price sensitivity to shipping costs, with 63% of cart abandonments by consumers in the United States being due to additional shipping costs [50]. Finally, in addition to their price sensitivity, consumers have high and increasing expectations with respect to the reliability and timing of deliveries [185, 208].

Given the high cost of last-mile deliveries in addition to the high expectations for timely and reliable deliveries, the analysis and improvement of last-mile logistics systems has become progressively more important. In particular, there has been an increased interest in innovative ideas to combat the inherent problems of last-mile deliveries [19]. The use of drones, for example, were investigated by DHL as a possible solution in combatting long delivery times [181], whereas other innovations include the implementation of drop-off points, where multiple consumers may collect their parcels [173]. An innovation of particular interest involves the use of crowd logistics, a business model that emerged alongside the rise of the collaborative economy.

The collaborative economy is a system by which individuals, businesses, and start-ups are encouraged to collaborate, with the aim of promoting access to resources and services, rather than the ownership thereof [31, 154]. This is enabled by cultivating an economic system that prioritises the availability and efficient utilisation of existing resources [2]. The need for such initiatives arises from the hyper-consumption characterising many modern-day societies, which has led to a severe underutilisation of assets, space, time, and trips. The European Environment agency found, for instance, that the average car occupancy across a number of European countries have declined to 1.45 passengers including the driver [79]. By creating platforms where various role-players may collaborate and share the excess of their existing resources, these inefficiencies are addressed and may be improved. The real-world implementations enabling a collaborative economy are broad, and the excess capacity in question might be in the form of extra rooms (*e.g.* AirBnB [4]), underutilised vehicles (*e.g.* Uber [194]), or excess time and cognitive capacity (*e.g.* Amazon's Mechanical Turk [8]). Attempts have been made to apply business models found in the collaborative economy as a basis to improve both the efficiency and sustainability of last-mile deliveries. In particular, the field of crowd logistics has shown promise as a solution to this vast problem.

Crowd logistics involves the use of ordinary people, as opposed to employed drivers, in the execution of logistics operations. It forms a system by which the demand of logistics operations is met by a supply of an informal crowd, that are willing to collaborate and utilise their excess capacity. Companies implementing crowd logistics make use of the Internet to offer a group of people (*i.e.* the crowd) the opportunity to perform logistic operations for a monetary incentive [163]. The crowd members, in turn, make use of their excess capacity (*e.g.* their spare time, pre-existing trips, an/or excess space in their vehicle) to perform these logistics operations [162]. Such a system may enable companies to act merely as facilitators of the logistics processes, enabling communication, data management, and payments, rather than making use of employed drivers or third-party logistics service providers [138]. Although the potential applications of crowd logistics are far-reaching, a large number of the existing real-world initiatives are focussed specifically on improving last-mile deliveries [38].

An exciting initiative within the field of crowd logistics involves the use of in-store customers to perform deliveries to online customers while on their way home from the store. This innovation aims to complement the existing last-mile delivery execution of companies that utilise an omnichannel sales approach (*i.e.* through an online sales platform in addition to their brick-and-mortar store). Attempts have been made by both Walmart and Amazon to incorporate such a system within their existing delivery methods, by incentivising their in-store customers to deliver orders to their online customers after visiting the brick-and-mortar store [21, 23]. Following these attempted implementations, this phenomenon caught the attention of various researchers. In particular, the theoretical and mathematical modelling of this initiative proved to be of interest. A mathematical formulation thereof was first proposed by Archetti et al. [9], as an extension to the classical *Vehicle Routing Problem* (VRP). Subsequently, a number of modelling approaches have been proposed towards this crowd logistics innovation, aiming to reflect the real-world considerations of such a system. Some of the most notable works include that of Dahle *et al.* [63], Gdowska and Pedroso [90], Dahle *et al.* [64], and Dayarian and Savelsbergh [66].

Although the proposed models vary in their naming conventions and specific implementation, the structure of the innovation remains the same — an omni-channel retailer utilises the occasional services of their in-store customers to assist their fleet of dedicated delivery vehicles in delivering the orders of a number of online customers. For the purpose of this thesis, this innovation is referred to as *customer crowd-shipping* and the associated in-store customers that occasionally perform deliveries are termed *occasional drivers* (ODs). Generally, the aim of these models are to provide routing solutions that minimise the total cost of serving online customers, comprising the cost incurred by the dedicated fleet of delivery vehicles and the compensation paid to ODs.

One of the primary challenges in modelling the customer crowd-shipping initiative is the design of an incentive scheme to adequately compensate ODs for their effort in delivering orders. A potential approach involves calculating the incentive based on the OD's required deviation as they deliver the order on their way home. If this is the only consideration in outsourcing an order, however, it may lead to instances where inexpensive orders are outsourced unnecessarily. This implies that the dedicated fleet of delivery vehicles could have served the online customer at a lower cost than the compensation paid to the OD. This introduces the additional challenge of determining which online customers should be outsourced to ODs. Embedded within this challenge is the notoriously difficult process of calculating the cost of serving a particular online customer using the dedicated delivery vehicle (*i.e.* marginal cost of insertion) [9].

An additional challenge relates to modelling the dynamic and stochastic behaviour of ODs within a customer crowd-shipping initiative. Unlike traditional last-mile delivery systems with dedicated delivery staff, customer crowd-shipping involves the use of in-store customers that act of their own accord. Given that ODs are not employed by the company in question, their behaviour are considered as outside of the direct control of the company. As such, rather than assigning an OD to a particular set of deliveries (as would be the case with the dedicated employees), ODs may be presented with a set of potential orders from which they may select an order to deliver along their way home. Each potential delivery may be associated with a monetary compensation, incentivising the OD to deliver the order. If a particular offer is sufficiently enticing to an OD, considered to be worth their time and effort delivering, the OD may accept the order. The OD's decision in considering the offered trips may be dependent on their individual socio-demographic characteristics, the additional effort required in delivering the order, as well as the incentive offered by the company. To model this decision-making process adequately, a modelling approach that reflects the autonomous nature of ODs, modelling them as independent agents, is required. Moreover, it may be necessary to quantify the effort extended by the OD to deliver an order, relating to the required deviation made on their way home from the retailer.

A potential approach towards modelling ODs as autonomous agents includes *agent-based modelling* (ABM). ABM is a modelling framework that follows a bottom-up approach to modelling complex systems with various autonomous and heterogeneous agents. Certain behaviours and rules may be imposed on the individual agents that exist in, and are influenced by, their environment. This modelling paradigm works well for systems involving dynamic decision making and social behaviour [27], making ABM a useful tool for analysing systems where various role-players are involved and have to make dynamic, autonomous decisions. In addition to ABM, consultation of the realms of microeconomics and transport economics may prove valuable.

1.2 Informal problem description

In light of the opportunity for research posed in the field of customer crowd-shipping, a dynamic and stochastic agent-based simulation model with the ability of capturing the workings of the customer crowd-shipping initiative is proposed. An adequate model encapsulating an incentive scheme, the autonomous decision making of ODs, and the selection of online customers for crowd-shipping, while providing insight into how these elements affect the key performance measurements of such a system, may prove valuable to the research and future real-world implementations of customer crowd-shipping.

1.3 Scope and objectives

The following objectives are pursued in this thesis:

I To *conduct* a comprehensive survey of the literature related to:

- (a) the e-commerce industry and the associated field of last-mile logistics,
- (b) the collaborative economy,
- (c) crowdsourcing and, in particular, crowd logistics,
- (d) the domain of classical VRPs,
- (e) computer simulation modelling with specific focus on ABM,
- (f) microeconomic theory and, in particular, consumer behaviour and the value of time, and
- (g) existing models of crowd logistics using ODs.

II To *formulate* a generic agent-based simulation model of a theoretical implementation of crowd logistics capturing:

- (a) the last-mile delivery system of a company with deliveries to online customers performed from a single depot,
- (b) the behaviour and decision making of ODs, and
- (c) an integrated last-mile delivery system where both the dedicated delivery vehicles and ODs perform deliveries.

III To *implement* the model with the ability to:

- (a) read input data pertaining the information of online customers,
- (b) provide the model user with a choice of VRP solution methodologies,
- (c) calculate incentives to be offered based on a user-defined incentive scheme,
- (d) dynamically calculate the cost-to-serve values of all customers to be served,
- (e) dynamically select customers for outsourcing, based on their cost-to-serve value, and

(f) determine a number key performance indicators (KPIs) and visually represent the performance of the system.

IV To *verify* the model iteratively throughout the development of the model using logical tests.

V To *validate* the model using techniques such as parameter variation, sensitivity analysis, and face validation.

VI To *evaluate* the agent-based simulation model under various operating conditions to analyse the performance for different scenarios.

VII To *recommend* sensible follow-up work related to the work in this study which may be pursued in future.

Customer crowd-shipping is an emerging field, both with respect to research as well as the real-world implementations thereof. To the knowledge of the author, no real-world data pertaining to an existing system are currently available. The model proposed in this thesis therefore aims to provide an abstraction of a theoretical customer crowd-shipping initiative. Thus, rather than modelling a particular real-world instantiation of customer crowd-shipping, the proposed simulation model aims to serve as a concept demonstrator for modelling such an initiative and to provide insight into various considerations of such a model. The input values and resulting output should therefore not be considered reflective of a real-world system. The aim is to identify trends and gain an understanding of such a system, which may be representative of customer crowd-shipping in general.

There are various crowd logistics models that utilise a crowd to perform logistics operations. The focus of this thesis, however, is specific to business-to-consumer (B2C) companies that utilise an omni-channel approach to sales within a set geographical region. The business therefore serves end consumers, either at a dedicated brick-and-mortar store, or by means of an online channel with accompanying doorstep deliveries. Moreover, it is assumed that deliveries originate from the brick-and-mortar store in question, serving customers in the vicinity.

1.4 Research methodology

The methodological approach taken in this thesis to design and develop an agent-based simulation model to investigate customer crowd-shipping is as follows:

I *Consult* the existing literature pertaining to the e-commerce industry and last-mile logistics, discovering the current lay of the land and challenges within the field.

II *Investigate* the existing literature pertaining to the collaborative economy, crowdsourcing, and crowd logistics as potential avenues for improving the last-mile deliveries of an omni-channel retailer.

III *Discover* the modelling techniques, tools, and approaches in the domains of VRPs, microeconomic theory, and agent-based simulation modelling.

IV *Develop* the required skills and knowledge pertaining to ABM, specifically using the AnyLogic simulation software environment.

V *Develop* the required skills and knowledge pertaining to VRPs, specifically using the Python programming language and the Pypeline custom library.

VI Iteratively *construct* a generic agent-based model that captures the various elements of last-mile deliveries.

VII Incrementally *develop* the model and add complexity in order to incorporate the use of ODs in the last-mile delivery system.

VIII Continuously *verify* the model throughout the development process to ensure proper functionality.

IX *Validate* the model by means of parameter variation, sensitivity analysis, and face validation.

X *Perform* an array of experiments, subjecting the model to various conditions and *analyse* the resulting performance of the model for different scenarios.

1.5 Thesis organisation

Apart from this introductory chapter, this thesis contains six chapters that document the methodological approach followed in fulfilling the objectives listed in §1.3. The organisation of these chapters are discussed subsequently.

Chapter 2 contains a literature review, detailing the various background considerations for customer crowd-shipping. This includes an overview of the e-commerce industry, as well as an in-depth discussion of the associated field of last-mile logistics. Thereafter, the collaborative economy is introduced and discussed. This includes discussions on the forces driving these modern business ventures, including a recent shift in consumer behaviour towards more collaborative lifestyles. A particular field within the collaborative economy, namely crowdsourcing, is discussed subsequently. This includes a framework for crowdsourcing models, that provides context for the remainder of the thesis. Finally, the field of crowd logistics is discussed in depth, detailing a number of existing crowd-logistics initiatives, before discussing the benefits, drawbacks, and risks associated with such initiatives.

Chapter 3 contains a review of the literature pertaining to a number of modelling techniques and approaches. In particular, this chapter documents the information required to develop an agent-based model for last-mile deliveries in the context of crowd logistics. First, a review of the literature pertaining to the classical CVRP is provided, detailing a number of the solution methodologies typically used to solve this well-known problem. Furthermore, an overview of computer simulation modelling is provided with a particular focus on ABM. Thereafter, an introduction of microeconomic theory is provided with reference to the modelling of consumer behaviour. The principles of microeconomics are subsequently applied in explaining the manner in which consumers value their travel time. As such, a discussion on the value of time is provided, detailing consumers' willingness to trade their travel time for money, as well as the associated influential factors. Finally, an in-depth review of existing crowd logistics models is provided. The models considered include implementations that correspond closely to the customer crowd-shipping scenario. The details of the existing models are elaborated upon, before a comparison of the various models is provided.

In Chapter 4, the development and implementation of an agent-based model of customer crowd-shipping is documented. The chapter opens with a brief description and motivation of the modelling and programming software chosen for the development of the simulation model. Thereafter a broad description of the proposed model is provided. This includes an in-depth description of the real-world initiative being modelled, a description of the required input data, as well as an overview of the assumptions and limitations of the model. Furthermore, the development of the agent-based model is documented in three phases. In the first phase, the approach to modelling a traditional e-commerce retailer serving a number of online customers from a single depot is detailed. A formulation of the capacity-constrained VRP, as well two well-known solution methodologies are detailed for the purpose of modelling the last-mile delivery system. The second phase documents the use of microeconomic principles in modelling the general behaviour and decision making of ODs. In particular, the modelling approach with respect to an OD's willingness to exchange travel time for money is documented. Finally, the third phase of model development documents the proposed integrated approach to customer crowd-shipping. This includes a description of the proposed algorithm that dynamically calculates the cost-to-serve values of all active online customers. The proposed incentive scheme is discussed, before detailing the proposed approach towards the selection of online customers to outsource. Finally, the third phase is concluded with a description of the decision making of ODs, relating to their willingness to accept proposed offers to deliver an order. Thereafter, the introduction of a number of KPIs measuring the success of the customer crowd-shipping model are documented. Finally, the development of a user interface is discussed, both with respect to the initiation of a simulation run and the subsequent real-time visualisation of the KPIs.

In Chapter 5, the verification of the agent-based model is documented. This includes the systematic verification of a number of key modelling aspects, such as the customer order frequency, the planning and execution of last-mile deliveries, the proposed cost-to-serve algorithm, the behaviour of ODs, as well as the calculation of delivery savings.

In Chapter 6, the evaluation of the agent-based model for customer crowd-shipping is documented. This includes the design of parameter variation and sensitivity analysis experiments, as well as the analysis of their respective results. Furthermore, the design of two scenario analyses are documented, and the associated results are discussed. Finally, the face validation performed by subject matter experts is documented.

The thesis is concluded in Chapter 7 with a summary of the content of the work performed. Additionally, an overview of the contributions made and an appraisal of the research performed in the thesis are documented. The chapter is concluded with suggestions for avenues of future research.

CHAPTER 2

Literature review

Contents

The aim of this chapter is to provide sufficient insight into the relevant domains that form part of customer crowd-shipping. In §2.1, a review of the e-commerce industry is provided, focussing on its associated benefits and challenges. This is followed in §2.2 by a discussion on last-mile logistics, with particular focus on its unique set of challenges, the drivers for improvement, and specific considerations for last-mile innovations. In §2.3 the collaborative economy is introduced and elaborated upon, with a particular focus on crowdsourcing in §2.4. Finally, a particular field of crowdsourcing, namely crowd logistics is discussed in §2.5.

2.1 E-Commerce

E-commerce is a term that refers to the use of the Internet for the purchase, sale, transport or trade of data, goods, or services [193]. The term may extend beyond mere buying and selling of goods to include business functions such as servicing of customers, e-learning, and electronic transactions.

An organisation may typically be classified as brick-and-mortar (*i.e.* physically based), virtual (*i.e.* purely e-commerce), or click-and-mortar (*i.e.* partially e-commerce). According to Turban *et al.* [193], the classification of an organisation is based on the nature of three constituent business functions, namely ordering and payments, order fulfilment, and delivery. If all of these functions are performed digitally, the organisation is considered a pure e-commerce organisation, whereas an organisation is classified as brick-and-mortar if all functions are performed physically. Finally, an organisation is considered partially e-commerce if only some of the relative functions are performed digitally.

E-commerce organisations may operate as a business-to-business (B2B), B2C, or consumer-to-consumer (C2C) platform. Online B2B transactions involve sales between businesses, such as a manufacturing plant purchasing parts from a supplier. According to Turban *et al.* [193], in 2018 roughly 85% of e-commerce transactions were B2B. In the same time period, 67.3% of all manufacturing shipments in the United States of America were performed by means of e-commerce. Although e-commerce was initially more useful for B2B transactions, the advances made in information and communications technology (ICT), as well as the major expansion of the Internet resulted in a rapid growth in B2C and C2C innovations [73]. B2C transactions involve customer-facing sales, such as end-consumers purchasing groceries online. C2C platforms, on the other hand, enable online transactions between consumers. Examples of such facilitating platforms include sites such as eBay and Gumtree [74, 102].

2.1.1 Background and history

The inception of the e-commerce industry can be tracked back to 1970 when communication systems were limited to rudimentary private networks [73]. These networks were generally limited for use by major corporations, the military, and academic institutes. According to Markoff [135], the first e-commerce transaction involved the sale of an undisclosed amount of marijuana. The transfer was made in 1971 between students from Stanford University and their peers at Massachusetts Institute of Technology, by means of a rudimentary network. After such a dubious inception, the e-commerce industry was quiet for a few years. In 1984, a company named CompuServe made an attempt at online retail by launching The Electronic Mall, which was targeted at a very limited group of home personal computer users [61]. The project granted users access to more than a hundred retailers online. Although it did not gain large commercial success, it was a forerunner for what was to come.

The development of the World Wide Web in the early 1990s allowed companies to have an online presence in the form of both text and photos. In 1992, the first e-commerce website targeted at end-consumers emerged in the form of an online book-store, named Book Stacks Unlimited [120]. After slow progress of the e-commerce industry, changes in regulation allowed commercial enterprises to operate *via* the Internet. This led to the founding of both Amazon and eBay, which paved the way towards e-commerce becoming a $150 billion industry by 1999 [187]. The wave of enthusiasm in the late 1990s resulted in a surge of investments in internet firms. A large number of these firms were over valued, however, relative to their intrinsic value. This ultimately led to the burst of the so-called *dotcom bubble*, as huge sell-offs ensued and the market crashed during the *dotcom bust* [105]. Consequently, the e-commerce landscape became challenging after 1999 with various B2B and B2C e-commerce companies declaring bankruptcy.

After this initial decline, the more resilient companies such as Amazon, eBay, and Wayfair held out and started exhibiting steady growth. The early 2000s was also the inception period of the Chinese e-commerce giant, Alibaba [143]. From 2009 onwards, major growth in commerce channels occurred over social networks [193]. The so-called social commerce channels include all commercial activities that occur on Facebook, Twitter, and other social networks.

The modern e-commerce industry has experienced growth beyond initial expectations. In 2012, global B2C sales exceeded the $1 trillion mark for the first time [75]. By 2015, more than 50% of the traditional retailers started incorporating an online sales platform, opting for a multiple-channel approach [139]. By 2019, global retail sales through e-commerce amounted to more than $3.5 trillion, accounting for 14.1% of all retail sales [51].

In 2020, the coronavirus pandemic, accompanied by the various global social distancing measures, exacerbated the growth in the e-commerce industry. Consumers shifted from physically visiting stores to purchasing basic goods online in unprecedented numbers [57]. Online retail sales in the United States of America grew by 32.4% from 2019 to 2020, exceeding $790 billion, accounting for more than 19.6% of all retail sales [5].

2.1.2 Benefits and challenges of e-commerce

As the e-commerce industry becomes more commonplace, a large number of manufacturers and retailers have supplemented their regular distribution channels with an online shopping experience. This addition to brick-and-mortar stores creates a shorter supply chain and allows for a more direct link to customers. Such strategies, referred to as omni-channel strategies, allows businesses to reach their customers by means of a variety of marketing and distribution channels [173]. The benefits of these developments may be viewed from the perspective of the customer, the company, and society as a whole.

The benefits resulting from incorporating e-commerce into a business are far-reaching. It grants companies a global reach at a reasonable cost and, since an online store is not limited to normal business hours, customers can be reached around-the-clock across the globe. The integration with different social networks and subsequent increase in data collection create opportunities for effective customer segmentation, customisation and marketing personalisation [193]. With improved ICT, the cost of procurement and communication, as well as costs relating to information storage, processing and distribution, also reduce significantly [187]. Furthermore, the accompanying supply chains benefit from increased coordination and collaboration, which allow for lower inventory levels and fewer delays [187, 193]. The decrease in physical distribution and the number of storage locations result in a pooling of risk, which further lowers the required inventory levels [185].

When considering the customer, it can be noted that the needs and priorities of customers may vary greatly across the various e-commerce business models. The customer of a B2B model, for instance, may have different expectations and experiences than that of the end-consumer in a B2C model. Many of the advantages experienced by the customers do, however, overlap. E-commerce sites often provide customers with a vast selection of vendors, products, and experiences, while allowing them to use comparison engines to find the best bargain [193]. Furthermore, customers have a greater level of flexibility, both with respect to when and where they shop, which are often coupled with the ability to customise their product on purchase. Finally, the convenience of doorstep deliveries that are often associated with e-commerce transactions is an additional benefit.

Society may also benefit from e-commerce and the accompanying movement away from brick-and-mortar stores. Doorstep deliveries may result in fewer trips to brick-and-mortar stores, ultimately resulting in a reduction in traffic. The economy primarily benefits by the means of more efficient firms, and improvements in productivity as a whole. According to Terzi [187], positive effects of the e-commerce industry on the economy can be seen in their contribution to economic growth, as well as the increased efficiency of labour and capital.

The limitations and challenges of e-commerce should, however, be considered alongside the benefits and opportunities. Many of these challenges are concerned with the level of customer satisfaction, which ultimately influences the customer spending and loyalty [149]. The challenges facing e-commerce may be viewed along four primary dimensions, namely economic, technological, social, and legal [24]. The social and legal dimensions include challenges concerned with the privacy, security, and accessibility of e-commerce. The challenges in the technical dimension are primarily concerned with the reliability, bandwidth availability, and integration capabilities of the ICT underlying e-commerce businesses. Finally, the economic dimension is concerned with the growth of e-commerce as a whole, with specific regard to the cost of e-commerce projects, infrastructure requirements, as well as the high cost and low speed of deliveries.

When considering B2C e-commerce companies in particular, the challenges faced in doorstep delivery speed are more pronounced. End-customer satisfaction is a particularly difficult challenge given the growing demands for the quality of service supplied, with a high demand for personalised, flexible experiences [197]. Additionally, delivering to end-consumers may be considered the most cost-intensive part of the supply chain [93]. Given the growing volume of goods for delivery, the high cost of deliveries, and the high expectations from consumers, the speed and quality of deliveries are considered major concerns. These challenges may be most profoundly experienced by companies that aim to incorporate a omni-channel approach to retail [170].

Amongst the companies aiming to incorporate an omni-channel approach, grocery retailers have been majorly impacted by the effects of e-commerce. A large portion of traditional grocery retailers has adopted a hybrid approach, offering both in-store sales, as well as online sales with accompanying deliveries. Additionally, existing online retailers (*e.g.* Amazon), utilise their existing knowledge and networks to incorporate online grocery shopping into their product offering [155]. Although it is expected that grocery shopping will contribute to the growth of the e-commerce industry, it requires the consideration of a number of unique challenges [6]. The perishable nature of a large portion of groceries needs to be considered when incorporating an online sales channel. Specifically, the delivery of such goods poses a major challenge [155]. Perishable goods need to be delivered directly to a consumer, or by means of a safe alternative such as refrigerated reception boxes [92]. The efficient management of the fulfilment process plays a pivotal role in the success of online groceries sales with respect to consumer adoption, business success and carbon footprint [88].

2.2 Last-mile logistics

The continued growth of the e-commerce industry is naturally followed by an increasing number of parcel deliveries [81]. Retailers have had an increasing online presence over the last few years, targetting the end-consumer. This includes both purely e-commerce retailers, as well as traditional brick-and-mortar retailers that are broadening their channels [139]. This shift towards online retail and its subsequent growth are largely driven by technological improvements, rising urbanisation, and an increasing buying power from the urban population [178]. It has increased both the quantity as well as the diversity of goods purchased, resulting in a higher demand for small and frequent deliveries [178].

As both the supply and demand within this industry grows, the need for efficient and effective logistics solutions become increasingly important. The lack of attention to the logistics associated with e-commerce previously led to the failure of a number of such ventures [185].

In general, the terminal node of a supply chain exists either as the traditional form of a brick-and-mortar store, or is executed as deliveries to consumers [92]. The *last mile* of a supply chain refers to this terminal stretch, where the product and the end-user meet. It is the part of the supply chain that physically connects businesses to their consumers, either directly at their homes or at a collection point [92]. The logistics concerned with this direct-to-consumer connection, including the planning thereof, have been studied in depth and is referred to as *last-mile logistics*. In Figure 2.1, different methods of delivering the end-product from the supplier storage location into the end-consumer's hands are shown.

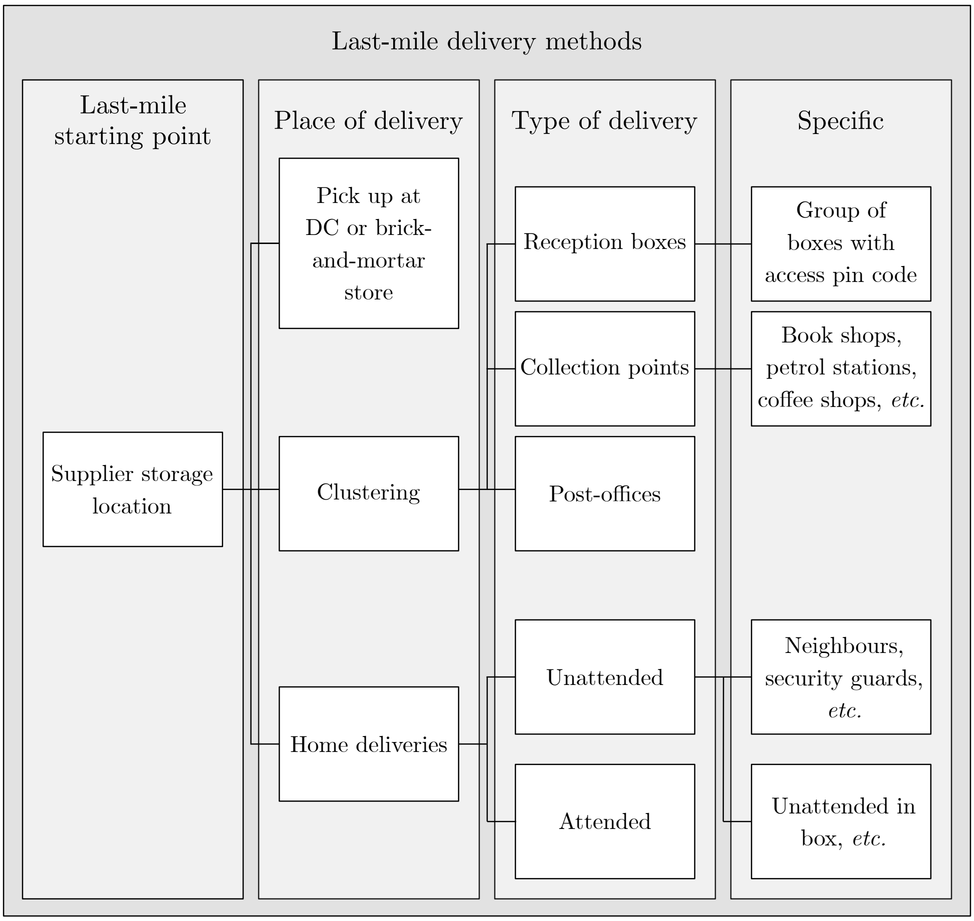


FIGURE 2.1: *Last-mile delivery methods adapted from Gevaers et al. [92].*

As noted in Figure 2.1, the storage location of the supplier is the origin from which last-mile deliveries are made. Traditionally, the consumer collects their product from the brick-and-mortar store or a distribution centre (DC). Although most retailers have operated in this manner over many years, it is being disrupted by new technologies and business models with alternative places of delivery through clustering or home deliveries. The clustering approach involves serving clusters of consumers at various points distributed throughout an area, such as post offices, collection points, or reception boxes. At post-offices, people have individual post-boxes which are embedded into a dedicated communal system. Collection points, on the other hand, use the premises of independent organisations to act as a bridge between the consumer and their package. The retailer delivers the aggregated orders of a given cluster to a specific independent organisation's location, such as a petrol station or book store, for collection by the consumer [92]. As an unattended albeit secure alternative, reception boxes are used. These are lockable containers which often include temperature control. A number of these secure boxes may be located in a central area in catering for a cluster of consumers. Once the delivery is made, consumers receive access codes to their respective boxes, which allow them to collect their packages at their convenience [118].

In addition to the traditional pick up and the clustering approaches, orders can be delivered directly to consumers' homes. Home deliveries can be performed either when the recipient is present (*i.e.* attended delivery) or in the absence of the recipient (*i.e.* unattended delivery). When making use of the latter, there may be a pre-arrangement with either neighbours or security guards to collect the package on the consumer's behalf. Alternatively, consumers may have a secure reception box at their home wherein groceries or high-value items may be delivered [118]. For items that are of less value, or in regions where petty crime is not an issue, packages may merely be left unattended at the front door of the recipient. In the case of attended deliveries the consumer has to physically be present to receive the package.

According to Olsson *et al.* [151], last-mile logistics refers to the process of planning, executing and controlling efficient and effective transportation and storage of goods from the placement of an order to the receipt by the end-customer. The multiple aspects that are embedded within last-mile logistics are captured in Figure 2.2.

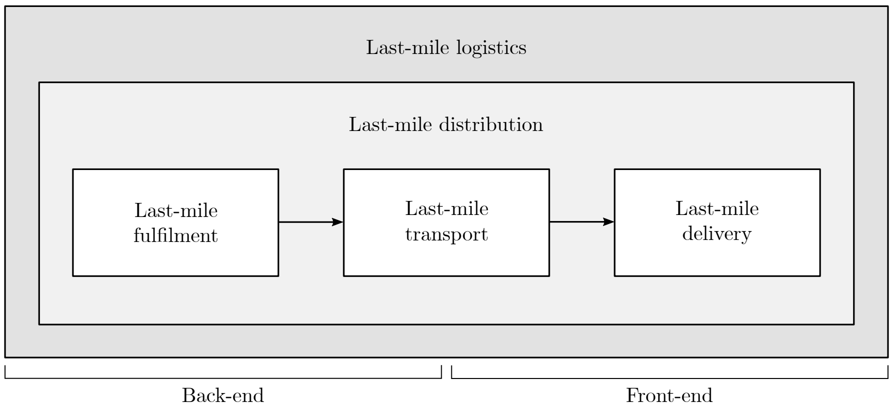


FIGURE 2.2: *A framework of last-mile logistics adapted from Olsson et al. [151].*

Within last-mile logistics, last-mile distribution concerns the handling, movement, and storage of goods for distribution to the end-user, as well as the channels through which the distribution is executed. It encapsulates a broad process stretching from the order fulfilment and transportation, up to the point where the end-user physically receives their order. Last-mile fulfilment is the process of preparing orders for delivery. This is followed by the last-mile transportation process whereby orders are transported to the terminal point of the supply chain. Finally, at the terminal destination, the last-mile delivery process ensures the completed physical delivery of orders to the end-users. Last-mile deliveries is seen as the customer-facing side of last-mile logistics.

Last-mile logistics is an encompassing term and, for the purpose of research performed in this thesis, the focus is on home-deliveries as well as the last-mile transport and last-mile delivery processes. The specific method of delivery, such as attended or unattended, as well as details regarding the storage of products, inter-organisation collaboration, and order fulfilment, are of less concern and therefore not explicitly considered. Furthermore, with the focus on B2C organisations, the customer in discussion refers to the end-consumer of the product. The remainder of this section considers challenges, opportunities and characteristics pertaining to last-mile logistics within this context.

2.2.1 Challenges and drivers in the context of home-deliveries

The so-called last mile is considered to be one of the least efficient, most expensive, and most polluting legs of a supply chain [92]. The typically fragmented and uncoordinated nature thereof further exacerbates traffic and pollution issues in urban areas. Improvements to last-mile logistics is therefore focused on creating a system where negative externalities are reduced, while continuing to provide an efficient service to consumers [164]. Various challenges concerning aspects such as customer expectation, time windows, cost, and sustainability are faced within the last mile of the B2C supply chain.

The first aspect of consideration is the rising consumer expectations of last-mile delivery quality and speed. Accompanied by the growth of the e-commerce industry, strong competition between role-players have resulted in deliveries of an exceptionally high standard. Furthermore, as companies prioritise the consumer's experience, increasingly more power is granted to consumers in dictating how the last mile should be performed [173]. As this becomes more commonplace, consumers may have greater expectations with respect to the convenience of deliveries.

Shorter delivery lead times, for example, improves the convenience for consumers as they have more control over when the order will arrive. As such, consumers are progressively more concerned about the speed of delivery [38]. The highly competitive e-commerce industry has made the option of same-day deliveries a common occurrence and, in some cases, decreasing the guaranteed delivery time to one hour [173]. This consumer convenience, however, comes at a trade-off for companies as the implementation of deliveries within set time windows results in longer distances travelled and higher costs [32, 118]. In addition to shorter lead times, there is an increased awareness around environmental issues. Consequently consumer decisions are based not only on the cost, convenience, and speed of the delivery, but also on the sustainability thereof. The expectation of decreasing carbon emissions is, however, not generally accompanied by a willingness to pay more for deliveries, or flexibility with respect to longer lead times [94].

A problem that often arises with attended deliveries is the absence of a recipient when attempting to perform the delivery. These attempted deliveries result in a driver having to return to specific customers multiple times, which results in accrued costs and wasted time [92]. This problem is countered by the inclusion of time windows wherein a customer can specify the time within which their delivery may take place. Specified time windows substantially decreases the chances of a failed delivery due to recipient-absence. Enforcing time windows into a delivery vehicle routing constraint, however, causes highly inefficient routes and results in a significant rise in distance travelled. As the length of the time window decreases, the distance travelled per customer increases [32, 178].

The cost of last-mile logistics forms a significant part of the supply chain, contributing up to 75% of the total cost [92]. Online retailers, however, often underestimate this cost and the importance of last mile efficiency [185]. The characteristics of the last mile has a direct influence on the cost of the system and are further discussed in §2.2.3.

In considering the cost of the last mile, customer density is of importance. A high customer density allows a single delivery vehicle to serve more customers for a shorter distance travelled. In many cases, the customer density is too low to justify the cost of delivery [92]. Performing home deliveries in areas of low customer density areas may result in vehicles with low load factors and uncoordinated routes. Such underutilisation of capacity and inefficient routes have significant effects on the cost of the system [164, 167].

In the pursuit of a more sustainable future, the environmental impact of the transportation industry as a whole has been a major topic of interest. According to the United States Environmental Protection Agency [195], the transport industry was responsible for roughly 29% of greenhouse gas emissions in the United States of America during 2019. Furthermore, while various industries seem to be making progress in reducing emissions, the transport industry is projected to increase emissions further in the years to come, due to the massive forecasted growth of the industry [164].

In a life cycle analysis, Hischier [109] evaluated and compared the environmental impacts of in-store and online shopping. It is concluded that the environmental impact cannot strictly be attributed to the form of purchase, but rather to travel behaviour of the consumer. For both in-store and online shopping, the key to reducing negative environmental externalities lies in optimising the accompanying transport activities [109]. In order to optimise these activities, the travel behaviour of consumers need to be considered and influenced. To affect change in consumers' travel behaviour is no simple task. According to Bjerkan *et al.* [25], personal travel behaviour is not impacted by e-commerce in itself, but rather by the manner in which the last-mile logistics is performed.

2.2.2 Opportunities and alternative delivery concepts

In recent years, a large focus has been placed on improving last-mile logistics, not merely by improving and optimising the current systems, but also by finding alternative methods to deploy. Considering the challenges described in §2.2.1, the general aim is to find solutions with a lower cost, while meeting the rising expectations of consumers and considering the sustainability thereof. Such opportunities may include horizontal and vertical collaboration, as well as crowd logistics. The impact of these opportunities within e-commerce and last-mile logistics is, however, largely influenced by consumers' openness to adopt the new innovations [131].

Collaboration between role-players within and across supply chains may contribute to the consolidation of last-mile deliveries. This current trend within last-mile logistics has the potential to mitigate the increase in traffic, while reducing transportation costs and carbon emissions through the improved utilisation of resources [147].

Vertical collaboration encourages the different role-players in a single supply chain (*i.e.* suppliers, manufacturers, logistics service providers and customers) to work together. Horizontal cooperation, on the other hand, requires companies on the same level of the supply chain, that would generally be seen as competitors, to work together [173]. Information sharing is a key element to collaboration, as it allows for more informed forecasts, leading to more effective planning and more efficient logistics management [185].

Crowd logistics is an initiative whereby certain logistics operations are outsourced to members of the public. An online platform is used to connect the supply of logistics tasks to willing crowd members. The application thereof in last-mile logistics has shown potential to disrupt traditional businesses [38]. There are economic, social and environmental benefits when crowd logistics is complemented to traditional deliveries. Through the utilisation of excess capacity from the public, the consolidation of orders are encouraged, resulting in reduced freight movements [173]. Finally, it aligns with the social trends of increased social and environmental responsibility, participation, and community-oriented living [31, 202]. This form of last-mile deliveries is further elaborated upon in §2.5.

2.2.3 Last-mile delivery innovation considerations

For the successful implementation of innovative solutions, it is essential that the last mile is conceptually understood. According to Gevaers *et al.* [92], the success of a last mile innovation is dependent on addressing five considerations, namely consumer service level, security and type of delivery, geographical area and market penetration, fleet and technology, and the environment.

The first consideration of a last-mile delivery innovation is the desired service level to the consumer. An increase in service level is typically met with the trade-off of increased complexity or increased cost. The primary elements of service level are the length of the guaranteed time-windows, the maximum guaranteed lead time, the frequency of deliveries, and the possibility of returned goods [92].

Furthermore, it is necessary to consider the method of home delivery and its associated security. Attended deliveries are regarded as safe, but has the prevalent issue of deliveries failing due to the recipient not being present to receive the delivery. Unattended deliveries, on the other hand, are highly dependent on the type of product, and carry higher security risks. Groceries, for example, may not stay fresh when left unattended for too long, whereas the risk of theft may be too significant when delivering high-value items.

Market penetration in a geographical area is especially important when it comes to delivering to a number of individuals. As the customer density increases, the average distance per delivery decreases [32]. Delivering to multiple customers in the same geographical region also allows for the consolidation of goods. In the case where an entire trip needs to be made for a single customer there is a significant increase in cost [94].

The type of delivery vehicles used for the last mile has an impact on the duration and cost of delivery. The fuel consumption, load capacity, difficulty in loading and unloading, as well as its ability to navigate traffic easily, impacts its effectiveness [92]. Various innovations prompted the use of electric vehicles, bicycles, autonomous vehicles, and drones for deliveries, with varied success [164]. Finally, the ICT employed has a major impact the efficiency of last-mile deliveries [30, 164]. The efficient flow of information can improve planning and routing decisions. Appropriate technology can result in saving time, money, and reducing the administrative load [92].

The primary impacts on the environment are as a result of inefficient deliveries. In some scenarios, deliveries of online orders may be more environmentally friendly than traditional shopping [92]. The emissions are, however, dependent on the above-mentioned characteristics of the last mile such as customer density, vehicle type and time windows.

2.3 The collaborative economy

> "Sharing has probably been the basic form of economic distribution in hominid societies for several hundred thousand years. It is based in human biological behaviours... and becomes a powerful force for social solidarity between communities" [158].

The collaborative economy is an economic system wherein ownership and access to resources are shared between role-players [154]. This technology-driven movement allows entities with an excess supply of goods and services to be matched with others that have a demand for those goods and services. A major distinction made between traditional seller-buyer systems and the collaborative economy is the shift from ownership to access, and from employment to collaboration [31]. Whereas in traditional systems, goods are produced to be bought and used by the customer as owner, the collaborative economy focuses on providing customers access to existing assets without ownership being required [31]. Similarly, where traditional systems focused on a fixed employer-employee relationship, the collaborative economy encourages freelancing, voluntary participation, and the utilisation of free time or unused capacity. The purpose is therefore to unlock the value in underutilised assets and excess capacity. In implementing such a system of sharing, individuals replace corporations at the centre of the economy and new market efficiencies are established [154, 174].

There are multiple business models that exist within this realm of the economy in an attempt to harness the major economic potential gains [156]. The most well-known and successful instantiations being Uber and Airbnb, affording customers access to transport and housing, respectively. Uber allows individuals to make their vehicles available for use during unused times, while allowing end-consumers access to transportation without having to own a vehicle. Uber drivers, in turn, have the opportunity to earn an income in their free time while working according to a flexible schedule [194]. Similarly, Airbnb creates a platform where individual home-owners may advertise available space and customers can find temporary accommodation that suits their needs [4].

Various terms, such as the *sharing economy*, *collaborative consumption*, *peer economy*, and *gig economy* are used to describe this economic model [31, 42, 136, 154]. Although these terms are often used interchangeably, the concept of collaboration seems to best capture the entirety of the movement. Sharing is based on the premise that people will behave appropriately, and does not require an equal return [158]. Collaboration, on the other hand, contains an element of reciprocity, whereby all parties involved stand to gain from the interaction. For the purpose of this thesis, the term collaborative economy is used to describe the entirety of this economic system.

2.3.1 Sharing and collaboration

Prince [158] describes sharing as a natural system of allocation and distribution. It has been prevalent in households and primitive societies for many years and is rooted in human bio-social behaviours. A problem arises, however, when sharing is applied to large and complex societies. Sharing and cooperation within societies are dependent on both reciprocity and strong linkages between individual members [31, 158]. It operates by mechanisms that are not governed by pure rationality, but also by social factors and emotions [158]. In societies where people tend to be individualistic, disconnected and motivated by self-interest, however, sharing becomes counter-intuitive [31]. These characteristics are epitomised by the modern-day, globalised, market-place economy with its large and complex distribution networks. For sharing and collaboration to become more prevalent, a number of characteristics need to be in place. One may require a better sense of community, more personal information about the individual, and recorded interactions over time [59].

Online platforms have been a solution to this problem of sharing. The collaborative platforms are often community-based, either with respect to proximity or purpose of the community, and typically require information about the individual upon registration. Finally, the actions and transactions of the individual may be recorded and utilised by the online system [31]. On the platform provided by Uber for example, passengers have the opportunity to rate their driver's service. Likewise, drivers may also rate the behaviour of the passengers after the completion of a trip. These ratings are captured and shared on the platform, aiming to cultivate trust between drivers and passengers alike. Such features contribute to building trust and encouraging collaboration by satisfying the demands of reciprocity and in-group linkages [154].

2.3.2 Consumption patterns

Veblen [199] coined the term *conspicuous consumption* in 1899 while referring to the manner in which the *nouveau riche* class spent their wealth. The purpose for which goods were bought and consumed leaned more towards self-aggrandising than towards utility. The reinforcing effects of marketing and materialism resulted in a surge of conspicuous consumption from the 1920s onwards [31]. In addition to the materialistic world-view that persuades individuals that they need to own a variety of goods in order to maintain the ideal lifestyle, marketing material continuously portrays the available goods to feed this lifestyle. These effects are further exacerbated by the evolution of easy payment systems. Various studies found that consumers tend to take less time, spend more money, and think less about their purchases when paying with credit cards as opposed to cash [80, 157, 184]. Finally, the shortening lifespan of goods have resulted in masses of seemingly durable products being prematurely discarded. The principle cause of this phenomenon is planned obsolescence, whereby goods are designed to have limited useful lifespans in order to drive up sales by forcing consumers to replace appliances on a regular basis [31]. These factors have all contributed to hyperconsumption experienced in the modern world.

In recent years, however, there have been winds of change. A steady shift in consumer values are becoming prevalent throughout society, triggered by the realisation that the limited resources on earth are not able to sustain the growth in consumption and waste. The millennial generation is especially conscious of such environmental issues and feel a responsibility towards contributing solutions [31].

2.3.3 Framework of collaborative business models

The change in consumption patterns and the movement towards collaborative consumption have led to a variety of business models springing to life. The broad scope of the collaborative economy allows for a large number of businesses and start-ups to be categorised under this umbrella-term. These include, amongst others, platforms for swap-trading and bartering, peer-to-peer currency exchange, crowdsourcing, shared workspaces, as well as car, bicycle, and ride sharing systems.

Botsman and Rogers [31] propose three categories by which collaborative consumption may occur, namely product service systems, redistribution markets, and collaborative lifestyles. Owyang [154] considers the collaborative economy from a business perspective with focus on the generation of value for businesses. This is encapsulated in a framework that outlines three different manners by which a company may generate value in the collaborative economy. It entails utilising a company as a service, motivating a marketplace where trade is facilitated, and finally, providing a platform for various collaborative activities. The framework aligns with the categories proposed by Botsman and Rogers [31]. The boundaries between the categories are not fixed and allows for business models to contain elements from multiple categories.

2.3.3.1 Product service systems

Ownership of a physical product is not necessarily required to gain access to the value of the product. In realising this, consumers are, for example, less interested in owning CDs or DVDs, but rather seek the music and videos that are carried on these products [31]. Product service systems are focused on offering access to the value of a product without the burden of owning the product.

The product service system correlates closely with the *company as a service* model proposed by Owyang [154]. This business model has been around for quite some time and forms part of the traditional economy. There are various familiar product service systems, such as car rentals, laundromats, hotels, and the renting of large-scale equipment, that rely on access rather than ownership. Technological advancements have, however, revolutionised this system and created opportunities for new models. Netflix, for example, requires a monthly subscription fee, which grants a user access to a wide variety of movies without the user owning the movie in a physical form. Products service systems are not limited to products from companies and may also incorporate privately-owned products to be shared or rented temporarily [31]. Companies such as Erento and Zilok facilitate such peer-to-peer collaboration [60, 76].

In addition to economic and societal benefits, the transformation of products into services has a profound environmental impact. It reduces the use of unrecoverable products which inevitably end up in land-fill sites. It has been estimated that the dematerialisation of music, for instance, could reduce the carbon footprint and energy use of the music distribution process by up to 80% [204].

2.3.3.2 Redistribution markets

A redistribution market encourages the reuse and resell of items, increasing their useful lifetime [31]. Platforms such as Gumtree, eBay, and craigs-list are some of the well-known redistribution markets [56, 74, 102]. In these models of the collaborative economy, second-hand items may be sold for a set cash price, be auctioned off to the highest bidder, or be traded for other items. Regardless of the form of transaction, the prolonging of the product life-cycle reduces waste and resources for new production [31].

Redistribution markets correlate closely with the *motivating a marketplace* model proposed by Owyang [154]. According to this model, the focus of companies is to foster a community where co-purchasing, reselling, and lending is encouraged. It allows the companies to extract value from products even past the point of first-sale by creating additional, unique customer experiences. This model is exemplified by the collaborative venture of eBay and the high-end outdoor clothing brand, Patagonia. By creating a dedicated online platform, the re-selling of second-hand Patagonia clothing products were encouraged and facilitated [182]. Patagonia benefited as the reduced prices of the second-hand products enabled their brand's reach into the lower-end of the market [154]. The re-sellers were able to extract more value from their purchases, while eBay increased the sales through their platform.

2.3.3.3 Collaborative lifestyles

Collaborative lifestyles go beyond the renting or sharing of merely products, and involves the exchange of intangible assets. Individuals may choose to collaborate by sharing their time, space, skills, and money [31]. It may take on many forms, such as shared working spaces, collaboration in innovations, crowdfunding initiatives, and crowd logistics. Collaborative lifestyle models often require a large degree of trust. Since it often involve human-to-human interaction, rather than a physical product, the social aspect is also more prevalent.

This mode of collaborative consumption correlates with the *provide a platform* model proposed by Owyang [154]. In this business model, a platform is provided for customers to collaborate through innovation and design, co-funding, co-distribution, co-marketing, and co-selling. This allows for stronger customer relations to form as customers are empowered to co-create business value. This may include crowdsourcing platforms such as Threadless, where T-shirt designs are co-created with customers, or crowd logistics, where customers may participate in some logistics operations [188].

2.3.4 Drivers and enablers of the collaborative economy

The collaborative economy is driven by a number of forces that work together in accelerating the growth of this industry. According to Owyang [154], the primary market forces which enable and drive the collaborative economy may be categorised as societal, economic, and technological drivers.

2.3.4.1 Societal drivers

The increasing population size coupled with the gravitation towards cities has led to an increase in population density. This enables smoother and more efficient forms of sharing, specifically when considering car-sharing initiatives [154].

As mentioned in §2.3.2, the drive towards more sustainable practises have resulted in the re-assessment of consumption patterns. This serves as a motivation to reuse, rent out or resell assets in order to extend their useful life [154]. The movement towards community-oriented living and the resurgence of local marketplaces are also contributing to the growth of the collaborative economy. There is a desire to move towards more simple and transparent forms of trade, wherein the individual on the other side of the transaction is known and trusted [31, 154].

Finally, there has been a generational shift towards more altruistic behaviours [31, 154]. The millennial and following generations have been raised largely in a world characterised by an abundance of wealth, as well as major income inequality and unsustainable business practises [174]. With the shift towards altruism, consumers are generally more welcoming to the collaborative economy due to the perceived sustainability of these business models [103]. This has led to a large push for an increased accessibility of goods and more sustainable practises. The collaborative economy has been a useful tool by which to achieve this aim.

2.3.4.2 Economic drivers

The monetisation of excess and idle assets is a major driver of the collaborative economy [154]. Resources that are idle for extensive parts of their useful lives provide opportunity for business. For example, Airbnb utilises empty rooms and houses, while Uber makes use of underutilised vehicles [136]. Individuals take this opportunity to use their excess capacity or idle assets in an effort towards financial independence [154]. In addition to physical assets, this includes the use of their cognitive capacity and skills to act as freelancers in contributing to crowdsourcing models.

As mentioned in §2.3.2, there has been a steady shift in consumer preference to access rather than ownership. The tendency of the current generation, specifically, is to break away from the high debt associated with ownership to a more sustainable consumption model [85]. The value of renting or leasing have been known by companies and consumers alike for many years [154]. The movement towards peer-to-peer renting, however, is more recent and have been implemented successfully by start-ups such as Erento and The Hire Hub [31].

2.3.4.3 Technological drivers

Technological advancements have enabled social networks and various online communities to support the collaborative economy [154]. Community-oriented lifestyle trends can extend beyond geographical communities and into the realm of social networks. The same familiarity, trust, and cooperation found within local communities may now be cultivated within online communities. This trend is exemplified by initiatives such as Couchsurfing, which is a global network from which one can connect with locals while travelling in a new city. In Couchsurfing, the local provides the traveller with a place to sleep for no charge, and often shares some local experiences around the city [55]. The level of trust that is present between strangers in such a situation is astounding, and is a result of technological advancement and verification systems in place [210]. This model, alongside many others, makes use of the user's digital reputation, which is established over time [154].

The advances made in mobile internet has further enabled massive growth in this industry. In an analysis performed by Martin [136], innovations within the collaborative economy typically depend on the utilisation of ICT. Similarly, Owyang [154] found that most sharing-oriented sites are focused on mobile applications which allow for access to use at any given time or place. Finally, advances in the ease-of-use and security of payment systems have been a major enabler for the collaborative economy. With traditional e-commerce sites leading the way, the smaller start-ups within this realm benefits from the advances in payment systems [154].

The effect of the societal, economic, and technological trends are reflected in the growth of the start-up scene within the collaborative economy. From the research conducted by Owyang [154] in 2013, the average funding of start-ups was in the range of $29 million, with a total of over $2 billion being invested in the 200 companies in the study. This is indicative of the acknowledged potential in this sector of the economy and the confidence in the nature of such business models.

2.4 Crowdsourcing

In recent years, a number of emerging business models make use of a large external crowd to perform tasks collaboratively. The term *crowdsourcing* was coined by Howe [112]. Although there are various definitions and a range of implementations which may differ in detail, there are certain common characteristics.

Generally, crowdsourcing starts with an individual, institution or organisation that requires a certain task to be completed. This is achieved by outsourcing the task, by means of an open call, to a group of individuals. The individuals may have varying knowledge and competencies, and should voluntarily chose to participate in the initiative [78].

The Internet acts as the main enabler for crowdsourcing, granting companies access to a seemingly limitless crowd. By collaborating with this crowd and tapping into their abilities, a company may enable the co-creation of value [95]. The crowd can be seen as a relatively inexpensive resource that has the potential of being continuously present, readily available, and providing fast reaction [127].

Hosseini *et al.* [111] propose a framework for a general crowdsourcing model. By analysing the literature on crowdsourcing, they deduced a taxonomy to bring a degree of order to a model which may become highly complex. This taxonomy provides a framework for researchers to untangle and describe the crowdsourcing model for a specific instance. It proposes that any crowdsourcing model can be viewed with regards to four pillars, namely the crowdsourcer, the crowd, the task, and the crowdsourcing platform. The hierarchical description of each of the pillars is provided, with each additional layer of the taxonomy becoming more instance-specific.

2.4.1 The crowdsourcer

The crowdsourcer refers to the entity that decides to utilise a crowd in completion of some required task. This entity may be an individual, company, institution or organisation. The key features of the crowdsourcer, as proposed by Hosseini *et al.* [111], relate to incentive provision, an open call, ethicality provision, and privacy provision.

1. *Incentive provision*
   The crowdsourcer may use incentives as a form of extrinsic motivation to encourage the crowd to perform the required task. The incentives mainly take one of three forms, namely financial incentives, social incentives, and entertainment incentives [111]. Financial motivations are the most prevalent in crowdsourcing models with platforms, such as Amazon's Mechanical Turk, Crowdflower, and oDesk, providing financial incentives in exchange for some task [177]. It has furthermore been shown that, with respect to extrinsic motivations within crowdsourcing models, financial incentives are preferred to social incentives [175]. The use of non-financial incentives, such as social recognition and entertainment value, have not been as effective in incentivising a crowd to perform a task [177]. Instances of these, however, exist and have been implemented to some degree of success [78].

2. *Open call*
   An open call means that the task, as specified by the crowdsourcer, is open for participation by an undefined crowd, as opposed to being assigned to a selected group [145, 192]. Although this implies that the choice to carry out the task ultimately lies with the individual crowd members, the crowdsourcer may have control over how wide they choose to cast their net. The specific crowdsourcing initiative may require that the open call be limited with respect characteristics of the crowd, such as their location or area, their skills, or the resources at their disposal. Thus the crowdsourcer may have a degree of control over who the task is proposed to, although they do not choose who eventually performs the task.

3. *Ethicality provision*
   Another feature to consider is the behaviour of the crowdsourcer, specifically with respect to ethics and morals. Three primary aspects are identified whereby the crowdsourcer should act in an ethical and compassionate manner [111]. First, there should be an opportunity for crowd members to opt-out of performing the task, even though this may have a negative effect on the crowd member and/or the crowdsourcer. Furthermore, it is advised that the crowdsourcer should provide feedback to the members of the crowd, informing them about the outcome of their task [133]. Finally, especially in the case of physical tasks, the crowdsourcer should ensure that the crowd members are not harmed while performing the task.

4. *Privacy provision*
   In certain crowdsourcing initiatives, the crowdsourcer may require personal information from the crowd members. The crowdsourcer has privacy obligations to ensure that the data of the registered crowd members are handled safely and securely. Information deemed personal should not be shared with other crowd members, companies or external entities [111].

2.4.2 The crowd

The group of individuals who are prompted to participate in the crowdsourcing task are referred to as the crowd. Hosseini *et al.* [111] define the important features of the crowd as diversity, undefined-ness, and suitability.

1. *Diversity*
   Diversity in the crowd is an important aspect to consider as it may contribute to the success of a crowdsourcing initiative. Although individuals may be diverse along multiple dimensions, different forms of diversity may have a greater or lesser impact on a specific

instance of the crowdsourcing model. The primary forms of diversity found to have an impact in crowdsourcing include spatial diversity [106, 108] and diversity in expertise [40, 104, 211]. To ensure that the desired level of diversity along these dimensions are reached, recruitment may assist in actively including individuals from different demographic profiles, geographic locations, and with varied experience or competence [111].

2. *Unknown-ness*
   Unknown-ness refers to the anonymity possessed by the crowd members within a crowdsourcing model. The first important aspect in this regard is the degree to which the crowd is unknown to the crowdsourcer [22, 172]. Furthermore, there is a degree of anonymity between the crowd members. People are therefore interacting and co-creating value with other members of the crowd without necessarily knowing the person with whom they are working [121]. Finally, when there is an end-user or customer within the crowdsourcing model, there may also be a degree of anonymity between the crowd members and the customer.

3. *Largeness*
   To ensure that the proposed task is adequately completed by the crowd, the crowd should be large enough. This refers to both a large number of crowd members, as well as the crowd being a comprehensive group of individuals with the capabilities to fulfil the requirements of the task [13, 40]. When the number of crowd members become too large, however, adverse effects, such as overload, confusion, and managerial complexity, may become too prevalent [3, 111].

4. *Undefined-ness*
   Undefined-ness refers largely to the selection procedure of the crowd. The degree to which the crowd is undefined rests on the set of criteria used when considering who may participate in the model. This may relate to geographic locations, abilities, qualifications, or workplaces of the potential crowd members [111]. On the other hand, a crowdsourcing model may be based on having a completely open call for anyone to participate [108, 145]. This aspect is synonymous to the open call feature of the crowdsourcer, but is from the perspective of the crowd [111].

5. *Suitability*
   The crowd may further be defined by their suitability to perform the required task. Suitability with respect to crowdsourcing has a number of dimensions, with varying degrees of importance, depending on the specific model. First, the competence of the crowd is considered as the members should have the basic skills required to perform the task [40]. Furthermore, the capacity to collaborate, which may include collaboration with other crowd members or with the crowdsourcer, is considered. Finally, the motivation of the crowd is considered, with a distinction made between intrinsic and extrinsic motivations. Extrinsic motivations, as discussed in §2.4.1, are provided by the crowdsourcer. Intrinsic motivation includes an element of mental satisfaction gained from engaging in the task, as well as the self-esteem gained in succeeding therein [78, 111]. It may also be seen as an opportunity for personal skills development and knowledge transfer [34, 86]. Crowd members may, for example, be intrinsically motivated merely out of a sense of love for the community to which they are contributing [99].

2.4.3 The task

The task is assigned by the crowdsourcer to be performed by the crowd. According to Hosseini *et al.* [111], the task may be described with respect to a set of elements which include the traditional operation, modularity, complexity, solvability, automation characteristics, user-driven nature, and contribution type.

1. *Traditional operation*
   First, the task may be described in the context of the traditional manner in which the business function would have been executed without the implementation of crowdsourcing. All tasks performed by means of crowdsourcing may traditionally have been performed by employees, or outsourced to an external contractor or company. The crowd therefore replaces the in-house employees or external entities when the traditional operation is replaced by the crowdsourced task [112].

2. *Modularity*
   Modularity refers to the concept of having a large or complex task, divided up into subtasks that are manageable by individual crowd-members [104, 111]. A modular task can, to some extent, be broken-down, standardised and made replicable. The resulting subtasks may then be easily assigned to individuals, who contribute to the greater task by executing their component. On the other hand, there are rare instances where a crowdsourcing task is an atomic task, outsourced in its entirety to a single crowd member.

3. *Complexity*
   Complexity refers to the quality of being multi-faceted and consisting of many interrelated parts. Crowdsourcing tasks may range from highly complex to more tedious tasks [146]. Although many models of crowdsourcing are for non-complex tasks, it has been shown that, in some cases, the crowd have an advantage over professionals in highly complex tasks [7, 192]. As an example, The National Aeronautics and Space Administration used the crowdsourcing platform Innocentive to find a solution to test Kevlar webbing under stringent conditions [113]. After professionals within the organisation attempted to find solutions for about three years, the highly complex problem was solved by non-expert crowd members.

4. *Solvability*
   Solvability refers to how easily the problem may be solved. In this regard, crowdsourcing tasks are often easily solvable by humans, but not by computers [69, 111]. The solvability of the task may influence the suitability of the crowd, specifically relating to their competence. Solvability differs from complexity and modularity, as a task may be complex and atomic (*i.e.* low modularity), but nonetheless be easy to perform for a human. As an example, delivering a package from a depot to a customer may be considered complex, as there are multiple interrelated considerations, and atomic as it cannot be broken down into standardised subtasks. It may, however, still be considered an easily solvable task for a human.

5. *Automation characteristics*
   Automation is defined as the integration of machines into a system that governs itself and performs tasks autonomously, without human intervention [101]. Tasks that are appropriate to a crowdsourcing approach are generally either difficult or expensive to automate [106]. The difficulty or high expense of automation is often the reason for utilising crowdsourcing, calling upon the skills or expertise of humans [111]. In the case that a

task is difficult to automate, but also not solvable by a crowd, crowdsourcing is not an appropriate solution.

6. *User driven*
A user-driven task is controlled and powered by users, whereby they are expected to use their own innovation or skills to complete the given task [70]. According to Hosseini *et al.* [111], crowdsourcing tasks are generally user driven and may fall into one of three categories, namely problem solving, innovation, or co-creation. A problem-solving task requires a solution to a given problem. Innovation tasks, on the other hand, requires the crowd to come up with original ideas or designs. Finally, co-creation involves the participation of the crowd to create a product or offer a service [201].

7. *Contribution type*
The contribution type refers to the manner in which a crowd member adds to the desired result of the crowdsourcing task. The contribution may be in the form of an individual contribution where the task is performed autonomously. On the other hand, the contribution may be of a collaborative nature wherein crowd members require one another and work together as a team in performing the task [111].

2.4.4 The crowdsourcing platform

The crowdsourcing platform connects the crowdsourcer with the crowd and assigns tasks by means of an open call. This platform is primarily online, allowing for the virtual connection of a vast number of entities. The main role of the platform is to act as an interface for the other constituent elements of the crowdsourcing model and allow for integration. Hosseini *et al.* [111] detail the interactions between the platform and humans, as well as the additional facilities of importance.

1. *Crowd-related interactions*
Crowd-related interactions refer to any event that requires interfacing between the platform and the crowd members. A great number of such interactions are possible and the details thereof are highly dependent on the specific implementation of the crowdsourcing model. A few core interactions are, however, present in a large number of crowdsourcing models and therefore worth mentioning. The platform should allow for the enrolment of a crowd member, which may include authentication and the declaration of their skills [111, 201]. The platform is therefore able to validate the suitability of the potential crowd member. Once a member has registered, the platform should be able to assign tasks to them, while providing assistance, coordination, and supervision, to ensure that the task is understood and executed [3, 58]. Finally, there should be a mechanism on the platform that allows crowd members to submit the outcome or output of their task, which may be followed by some form of feedback from the crowdsourcer or relevant party.

2. *Crowdsourcer-related interactions*
The crowdsourcer-related interactions involve the interfacing between the platform and the crowdsourcer. The crowdsourcer should be able to provide a task broadcast by which an open call to work is made to the appropriate crowd members [111]. In some cases, there may be a need for negotiations with respect to the price of incentives and/or the specified deadlines [84]. Furthermore, the crowdsourcer should have a mechanism to verify results and provide feedback to crowd members.

3. *Task-related facilities*
   The platform should provide certain facilities for the task being performed. First, certain tasks may have a minimum requirement of quality or quantity. The platform should be able to provide such thresholds [33]. Furthermore, it should have the capability of storing the history with the associated data of completed tasks. Finally, the platform should be able to aggregate and display the progression and outcome of tasks as they are completed.

4. *Platform-related facilities*
   The platform-related facilities are important aspects relating specifically to the platform. These may include mechanisms to manage misuse of the platform, either by the crowd or the crowdsourcer. Furthermore, the ease of use and attractiveness of the platform should be considered [3]. Finally, the platform should be able to provide some payment mechanism by which crowd members are paid for completed tasks.

2.5 Crowd logistics

Crowd logistics is a subclass of crowdsourcing, aimed at solving problems concerned with logistics. The crowdsourcer provides an open call for the crowd to perform logistics operations, such as the transport or storage of goods. The crowd members, in turn, meet this call by using their pre-existing trips, or excess storage capacity to perform the task at hand. Thus, the demand of logistics operations is met by the supply of excess capacity in exchange for an agreed compensation [145, 162].

It forms part of the greater collaborative economy movement, as crowd members choose to share their excess time, space, and skills in collaboration with the crowdsourcer. In particular, this form of co-distribution may be considered a collaborative lifestyle model, as value is unlocked through the sharing of both physical and intangible assets [31, 162]. As with other areas of the collaborative economy, it has been popular among practitioners, with a large number of related start-ups arising over recent years [160].

In this form of crowdsourcing, the geographic locations of both the customers and the crowd members are important. Given that physical elements, such as products, vehicles, and storage space, are core to crowd logistics tasks, the possible instances of collaboration are limited to people within a shared geographical area. Data and computing technologies may therefore be critical in connecting the crowdsourcer to the correct crowd members in serving the relevant customers [145].

2.5.1 Core elements in crowd logistics

Rai *et al.* [162] define crowd logistics as "an information connectivity enabled marketplace concept that matches supply and demand for logistics services with an undefined and external crowd that has free capacity with regards to time and/or space, participates on a voluntary basis and is compensated accordingly." This definition is based on a number of aspects that are deemed to be crucial in understanding a crowd logistics initiative. The core elements identified, namely compensation, voluntary participation, external and undefined crowd members, the crowd network, free capacity, and technological infrastructure, are considered with reference to the crowdsourcing framework as discussed in §2.4.

First, the crowdsourcer provides a supply of logistics operations to be performed by the crowd. A critical aspect is that there should be a form of compensation offered by the crowdsourcer.

Given that the crowd logistics initiative may improve the overall service level to customers, the compensation offered should reflect the value added by the crowd [162].

Furthermore, there is generally no hiring process and the crowd remains external to the company. As such, participation is generally on a voluntary basis as opposed to having work assigned [144]. The platform typically provides the crowd member with a list of geographically proximate operations from which they can choose which to undertake [41]. The undefined nature of the crowd may, however, be a source of stress for practitioners as it is difficult to guarantee quality and service.

The task performed by the crowd may be any logistics-related activity. Some initiatives allow individuals to rent out space in their garages as a warehousing service [162]. Crowd members may furthermore use public transport to deliver parcels. In order for any of these logistics operations to be economically, socially, and environmentally beneficial, it should be performed strictly by utilising free capacity. This may refer to pre-existing trips, free time, or excess storage space [202].

The crowdsourcing platform, in this case referred to as technological infrastructure, is key to coordinating the demand and supply for logistics providers. It is the hub by which business functions are executed and enables the flow of information, goods, and finances [138].

2.5.2 Existing crowd logistics initiatives

Although various logistics operations may form part of crowd logistics, the greatest potential for disruption lies among businesses that perform last-mile deliveries [38]. This includes the last-mile trip of e-commerce sites, or local deliveries from brick-and mortar retailers. This form of crowd logistics, commonly referred to as crowd-shipping, has been growing in popularity both on an academic front and in real-world implementations. Various start-ups and experiments by larger corporations have been initiated over the last few years to implement crowd-shipping [14, 21, 37, 68, 130, 148].

Myways is an initiative of DHL in Sweden [130]. It is aimed at improving the last-mile system of DHL by opening the final delivery leg to crowd-shipping. Online customers indicate the time window in which they would prefer to receive a package. Once a package is available for delivery, willing drivers are notified that it may be picked up at a collection point. These part-time drivers may then notify the system that they have accepted the delivery request, pick-up the package at the collection point, and deliver it to the awaiting online customer. This initiative has been found to be popular among students who aim to earn some additional money while being able to work flexible hours.

In Belgium, the postal operation provider *Bpost* ran an experimental initiative with an application named *bringr* [37]. The application connects individuals who want to send parcels with willing drivers. *Bpost* merely acts as a facilitator in the process, collecting a percentage of the compensation paid to the driver. *Postmates*, *Instacart*, and *Algel*, on the other hand, are companies focusing on delivering food, groceries and other household items [68]. By utilising a crowd of private drivers, the platform connects local stores to their consumers in urban areas.

Another interesting instance of crowd logistics is the Mumbai lunch box delivery system. A large network of lunch box carriers (*dabbawalas*) utilises the public transport system in Mumbai to deliver home-made lunches across the city. At any moment in time, up to 5 000 *dabbawalas* are operational in the system, with each serving up to 30 customers [14]. They collaborate extensively to pick up, sort, and deliver the lunch boxes using only the public transport network.

Walmart launched an initiative by which deliveries to online customers are outsourced to a crowd. The crowd, however, is not completely undefined in this instance, but rather consists of their in-store customers. Once an in-store customer completes their shopping, they may opt to deliver to an online customer that is along their route [21]. This model was later adapted, allowing start-ups, such as *Lyft* and *Deliv*, to also perform deliveries [148].

2.5.3 Benefits

According to Rai *et al.* [162], the benefits of implementing crowd logistics can be viewed from three perspectives, namely economic, social, and environmental.

From an economic perspective, crowd logistics may benefit businesses, crowd members, and customers alike. Benefits to customers may be in the form of access to a more extensive range of products, deliveries that are faster and more traceable, and service that is more flexible, convenient, and affordable [12, 138, 145]. Businesses, on their part, may benefit largely as crowd logistics models allow for a reduction in transportation costs by use of a flexible workforce with low initial investment required [38, 145]. Finally, crowd members may earn additional income through personalised and flexible opportunities to participate in such models.

Furthermore, the social benefits are primarily realised through the collaboration with a local customer network which may create a sense of community. The increased level of collaboration that is enabled, fosters a community-relationship between companies and their customers, as well as between crowd members and their neighbourhoods [145, 162]. The social benefits gained by crowd members are thus the satisfaction of intrinsic motivations, such as the desire to contribute in their community, the desire for variation in life, and the desire to contribute to sustainable practises [145].

Finally, the environmental benefits of crowd logistics are seen as the movement's most significant contribution [162]. As it incorporates the use of pre-existing trips and utilises loading space more efficiently, multiple trips may be consolidated [47]. The resulting routes are more efficient, cause less traffic and, as a consequence, contribute less to pollution [145].

2.5.4 Drawbacks and risks

Despite the clear benefits gained from crowd logistics, there are some inherent risks and drawbacks associated with such models. Mladenow *et al.* [145] provide reference to some of the main challenges, namely the distribution of responsibilities, additional costs, delivery delays, as well as security and privacy issues. The handling of physical objects inherently carries the risk of it being damaged or lost. In such scenarios, the distribution of responsibilities becomes an issue with the main concern being who shall be held liable for the losses [202]. Insurance of shipments could be a potential solution, but serves as an additional cost that needs to be carried by a party in the system. Although crowd logistics initiatives are aimed at saving transport costs, additional costs such as these may still arise. Other costs may be incurred through software development, additional training to customers, new packaging for such special deliveries and the like. Furthermore, the reliability and trustworthiness of crowd members should be considered. If this is not monitored closely, the timeliness of deliveries and, in severe cases, the safety of customers, may be compromised. As such, customers may be averse to the idea of providing their home address to random members of the crowd. Given these risks and the increasingly strict regulations with respect to data protection, the final drawback relates to maintaining the privacy of customers [202].

2.6 Chapter summary

An overview of the e-commerce industry was provided in §2.1, detailing its background and history, as well as its associated benefits and challenges. Thereafter, the notion of last-mile logistics was discussed in §2.2 with reference to its associated challenges and drivers, the opportunities for development, and the considerations of last-mile innovations. The collaborative economy was introduced in §2.3, with particular focus on the evolution of sharing and consumption, the various categorisations of collaborative business models, as well as its various drivers. Penultimately, crowdsourcing was elaborated upon in §2.4, including descriptions of its constituent elements, namely the crowdsourcer, the crowd, the task, and the crowdsourcing platform. Finally, crowd logistics was discussed in §2.5, detailing its core elements, describing various instances of crowd logistics, and finally providing a number of associated benefits, drawbacks, and risks.

CHAPTER 3

Modelling prerequisites

Contents

This chapter aims to provide the reader with the necessary background to the various fields considered in the remainder of the thesis. In §3.1, the well-known capacity-constrained VRP is introduced, and a number of its solution methodologies are considered. Thereafter, simulation modelling is introduced in §3.2, with particular focus on ABM. Furthermore, in §3.3, microeconomic theory and consumer behaviour are discussed. This is followed in §3.4 by a discussion on the value of travel time and the manner in which consumers are willing to trade time for money. Finally, existing crowd logistics models from literature are discussed in §3.5.

3.1 Capacitated vehicle routing problem

The utilisation of computerised algorithms in the planning of routes has become more prevalent over the last few decades. This is partly due to an increase in real-world demands as a consequence of the expanding world-population and the growing global economy. It is also driven by technological developments leading to increased computational power. This has enabled savings in both the planning and distribution processes of supply chains [189]. These computerised algorithms and software programs find their mathematical underpinning in the VRP.

The VRP involves determining a set of delivery routes to a number of customers, which is executed in multiple trips. A trip starts and terminates at one or more depots, and follows an appropriate road network. These trips may be performed by multiple vehicles. The aim is to obtain a set of routes to follow that minimises the cost of operations, satisfies customer demand, and adheres to the constraints imposed in the particular instance of the problem [189]. The problem was initially formulated by Dantzig and Ramser [65] in 1959 in an attempt to optimally route a fleet of vehicles delivering gasoline from one depot to a number of service stations. The mathematical model proposed was in essence a generalisation of the well-known

Travelling Salesman Problem (TSP)[1]. The TSP is generalised to the VRP by taking into account that customers may have a certain demand and delivery vehicles may have set capacities. The possibility that the global demand might not be satisfied in a single trip is therefore considered and multiple routes may need to be determined.

In their founding paper, *The truck dispatching problem*, Dantzig and Ramser [65] proposed an algorithm to find a near-optimal solution to the routing of multiple trucks delivering gasoline to a number of fuel stations. This algorithm, however, is only applicable to smaller instances of the VRP and more efficient algorithms are required for larger, real-life instances. Clarke and Wright [46] proposed a more efficient, greedy algorithm which gained popularity as it could solve larger instances while attaining near-optimal solutions.

Equipped with a simple and efficient algorithm to solve the problem, a myriad of implementations and variations on the theme of VRPs were developed by academics and practitioners alike. The least complex and most studied variant is the *Capacitated VRP* (CVRP) which is considered the basis variant of VRPs [189]. In the CVRP, multiple delivery vehicles with identical capacities are routed from a single, central depot to fully satisfy a deterministic customer demand. The aim is to minimise the total distance travelled while respecting the road network and adhering to the capacity constraints.

A multitude of variants arose from the basis of the CVRP, aiming to either incorporate additional real-world considerations or extending the CVRP to new use-cases. The most well-known and studied variants, as mentioned by Toth and Vigo [189], are shown in Figure 3.1. The *Distance-Constrained VRP* (DCVRP), for instance, sets a maximum allowable distance or time for each vehicle to travel. This may be applicable when drivers have limited time in which to perform deliveries. In the *VRP with Time Windows* (VRPTW), a time interval is assigned to each customer wherein the delivery should be conducted. Time windows may be added as an additional constraint in the application of other variants as well, extending the problem when required. In the *VRP with Backhauls* (VRPB), the case is considered in which there are customers from which a pick-up is made. In this problem, vehicles visit regular customers first to perform deliveries, after which backhaul customers are visited to pick up products. A further extension of this variant is the *VRP with Pickup and Delivery* (VRPPD). In this variant, a customer may have both a demand for delivery and pickup. Throughout the execution of a route, pickups and deliveries may be performed as required. Time windows constraints may be incorporated into the VRPB and VRPPD to form additional variants of the CVRP, such as the *VRPB with Time Windows* (VRPBTW) and the *VRPPD with Time Windows* (VRPPDTW).

3.1.1 Problem definition and notation

For an instance of the CVRP, it is considered that there is a set of n customers $N = \{1, 2, \ldots, n\}$ to be served from a single depot, indexed as point 0. The demand of customer i is denoted as q_i, where $q_i > 0$. The set of k homogeneous vehicles $K = \{1, 2, \ldots, k\}$ are responsible for delivering to these customers. All vehicles have identical capacities of $Q > 0$ and operate at an identical speed and operational cost. For a vehicle travelling from point i to point j, a travel cost of c_{ij} is incurred.

Solution methodologies to the CVRP and its variants have been extensively researched from the 1960s, with a myriad of approaches following combinatorial optimisation techniques proposed in literature. The classification of these techniques is a complicated task due to the diversity

[1]In a TSP, a travelling salesman aims to find the shortest route to travel from a depot, and to visit each customer in a list of m customers exactly once, before returning to the depot.

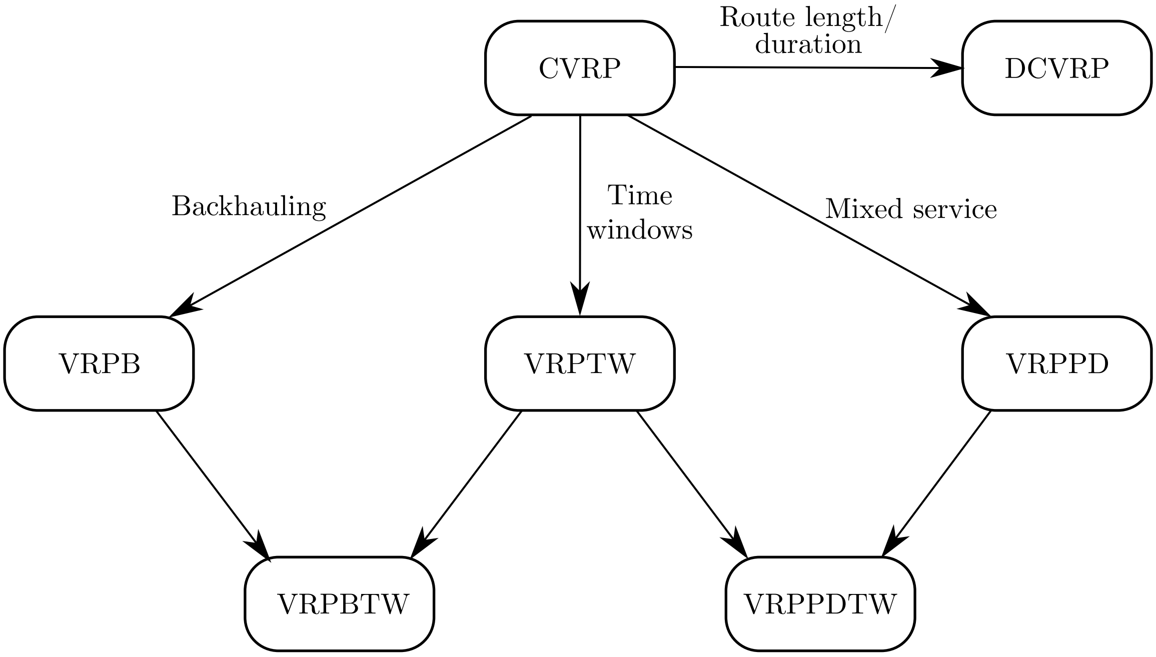


Figure 3.1: *Basic variants of the VRP and their interconnections adapted from Toth and Vigo [189].*

and intricacy of the algorithms [53]. Broad classification schemes focused on the core ideas may, however, be instructive. The approaches can be broadly categorised into three forms of solution, namely exact solution approaches, classical heuristics, and metaheuristics. A categorisation of the various available approaches discussed in the remainder of the section are shown in Figure 3.2.

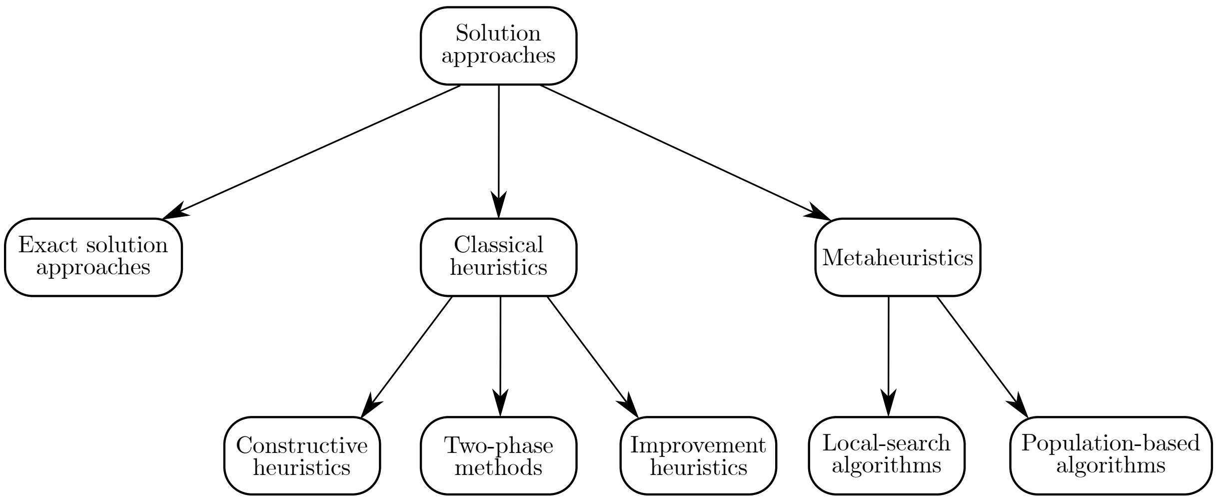


Figure 3.2: *The categorisation of CVRP solution approaches.*

3.1.2 Exact solution approaches

There are various methods to solve the CVRP exactly and obtain an optimal solution. These algorithms include branch-and-bound, branch-and-cut, dynamic programming, set partitioning and, most recently, branch-and-cut-and-price. Many of the exact methods originate from the study of the TSP and have been extended to the VRP in an attempt to find optimal answers [191].

Major advances were initially made in this respect with the use of branch-and-bound algorithms, originally proposed by Christofides and Eilon [44]. The VRP may be formulated as an integer programming problem after which the branch-and-bound method is employed to find an optimal solution. This approach is based on the idea that all feasible solutions to the problem can be partitioned into subsets of solutions. Thereafter, multiple potential solutions may be eliminated by solving a single sub-problem [206]. By incorporating the use of the *cutting plane method*, the branch-and-bound method may be extended into the branch-and-cut method. The use of a cutting plane, either at the top of the tree or at each node, may reduce the size of the tree significantly. This in turn accelerates the process of attaining an optimal solution [142]. These methods have been extensively used in solving the CVRP, with reviews on the progress being studied extensively by Toth and Vigo [189, 190, 191].

These methods, however, are not easily scalable and are rarely useful in practical applications [54]. This is due to the hard nature of this combinatorial optimisation problem and the resulting computational burden [54]. In 2002, Cordeau [54] reported that only relatively small instances of up to 50 customers can consistently be solved to optimality. In a comprehensive review of VRP literature in 2009, Laporte [125] stated that the best exact algorithms of the time are able to solve instances involving roughly 100 customers. Similarly, in 2012, Baldacci *et al.* [17] reported that recent improvements to exact algorithms for the VRP enabled the optimal solution for instances of up to 100 customers. Finally, Toth and Vigo [191] reported in 2014 that improvements in exact methods have enabled the consistent discovery of optimal solutions for instances of up to 200 customers. Although this improvement is significant and an impressive feat, many real-world instances exceed this number of customers. Furthermore, real-world applications often require an instantaneous solution, whereas exact methods may take long to compute.

The benefit of finding optimal solutions are often outweighed by the computational burden and the associated time taken to attain such solutions. As such, methods have been developed to enable the trade-off between optimality and computational time. By implementing heuristic approaches, optimal answers are sacrificed for the benefit of faster implementation.

3.1.3 Classical heuristics

The need for quality solutions within a reasonably short timeframe drove the development of heuristic methods to solve the CVRP. These techniques, referred to as classical heuristics, perform a limited search of the solution space, not allowing for the deterioration of the objective function [125]. They are highly adaptable and can easily incorporate various real-life constraints while producing quality solutions within modest computing times. These characteristics have made classical heuristics extremely popular both academically and commercially [124]. According to Toth and Vigo [189], the classical heuristics can be broadly categorised into three categories, namely constructive heuristics, two-phase methods, and improvement heuristics. The remainder of this section is dedicated to providing detail on some of the more prominent examples of classical heuristics.

3.1.3.1 Constructive heuristics

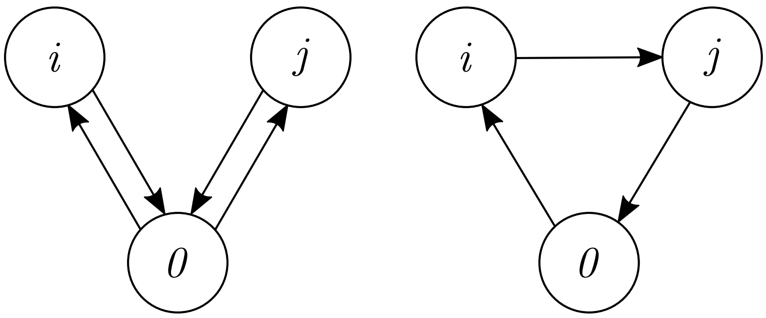
Constructive heuristics operate by iteratively constructing feasible routes, either by inserting a customer into a route, or by merging existing routes. This is generally done in a greedy manner. As the route is constructed, a set of definitive decisions are made while continuously considering the cost of the solution [200].

The savings algorithm, proposed by Clarke and Wright [46] in 1964, is one of the most well-known methods for solving large-scale VRPs. This constructive algorithm utilises the idea of savings incurred by connecting two customers on a route. It works in the following manner:

Step 1 Construct an initial solution of n routes, directly linking the depot to all customers. Calculate the savings incurred when linking customer i to customer j in a route as $s_{ij} = c_{i0} + c_{j0} - c_{ij}$ for $i, j = 1, \ldots, n$ and $i \neq j$, where 0 denotes the depot. Rank the savings s_{ij} and list it in a non-increasing manner.

Step 2 The second step may be performed using one of two approaches:

- The parallel version (best feasible merge): Run through the list of savings sequentially. For each savings value s_{ij}, determine if there exists two routes, one containing the path from the depot to j $(0, j)$, and one containing the path from i to the depot $(i, 0)$, that can be feasibly merged. If these can be feasibly merged, remove the paths $(0, j)$ and $(i, 0)$ and introduce the path (i, j). This is illustrated in Figure 3.3, with the initial separate routes shown in Figure 3.3(a) and the newly merged route shown in Figure 3.3(b).
- The sequential version (route extension): Run through the list of routes sequentially. For a current route $(0, i, \ldots, j, 0)$, determine a savings value s_{ki} or s_{jl} that can feasibly be used to link the current route to a separate route containing either the path $(k, 0)$ or the path $(0, l)$. Once a feasible merge is made, continue with the current route, repeating the procedure. Once no feasible merges can be made to the current route, continue to the next route and repeat the procedure. The algorithm ends when no feasible merges remain.



(a) *Routes before merge* (b) *Routes after merge*

FIGURE 3.3: *An illustration of a route merger in the Clarke and Wright savings method.*

It was shown by Toth and Vigo [189] that the parallel version outperforms the sequential version, producing better results in a smaller amount of time.

According to Vidal *et al.* [200], the savings algorithm was improved over time, most notably by accounting for a flaw embedded in the original method. Given the nature of the algorithm and its focus on performing merges with large savings first, the flaw lies in that good routes are produced initially, with less-efficient routes produced near the end of the algorithm [189]. The contributions by both Gaskell [87] and Yellow [209] addressed this tendency, specifically in avoiding the creation of circular routes. A shape parameter λ was introduced, changing the savings calculation to $s_{ij} = c_{i0} + c_{j0} - \lambda c_{ij}$. By introducing $\lambda \geq 0$, the importance of the distance between the vertices to be joined can be controlled. A larger value of λ enhances the importance of the distance between the vertices to be joined. It is found that values of λ between 0.4 and 1 produce good results [98].

Further improvements were made with a focus on reducing the computational time for calculating, sorting and storing savings. Due to the massive advances in modern-day computing power these improvements are becoming decreasingly justified [124].

3.1.3.2 Two-phase methods

In two-phase methods, the CVRP is split into two natural phases, namely the determination of feasible clusters and route construction. In *cluster first, route second* heuristics, some clustering[2] technique is used to generate clusters of customers, each associated with the route of a single vehicle. By implementing capacitated clustering techniques, it is ensured that the cumulative demand of customers in a single cluster does not exceed the associated vehicle capacity. As such, in the second phase of the heuristic, each cluster may be treated as a TSP. On the other hand, in *route first, cluster second* heuristics all customers are initially added to a single cycle, which is treated as a TSP. Thereafter, this giant tour is cut into several feasible routes, starting and ending at the depot [200]. The remainder of this section details some implementations of two-phased methods, with particular focus on cluster first, route second approaches.

Sweep algorithm

The sweep algorithm was proposed by Gillet *et al.* [96] in an attempt to obtain solutions to large instances of the CVRP. Routes are constructed in a cluster first, route second approach, with clusters formed by means of a rotating ray centred around the depot. Thereafter, the route of each individual cluster is optimised.

In the most simple implementation of the sweep algorithm, each customer i is represented by their polar coordinates (θ_i, ρ_i), where θ_i and ρ_i represent the angle and distance from the depot, respectively. Starting at an arbitrary customer, all customers are sorted according to their angle around the depot.

After this initialisation, feasible clusters of customers are formed and added to the associated delivery vehicles. For the initial vehicle k, with a capacity of Q, the customer with the smallest angle is selected and added to the associated cluster. By selecting the next customer based on increasing angles, customers are added to the cluster so long as the capacity constraint of the associated vehicle is not violated. Once the capacity of vehicle k has been exceeded, an additional cluster is added, corresponding to vehicle $k+1$, and the sweep continues. In essence, starting at an arbitrary customer, the first phase sweep around the depot, adding customers to vehicles so long as the capacity constraints are not violated.

In the second phase, each cluster obtained is treated as an individual route, starting and ending at the depot. Since each route is allocated to a single vehicle and its capacity has already been considered, the resulting optimisation of the route may be treated as a TSP. This sub-problem may be solved by heuristic approaches or by exact methods, depending on the application of the problem [124].

The speed and scalability of the sweep algorithm are attractive features. For a fixed number of customers per route, its computational time increases linearly with an increase in the total

[2] Clustering is the problem of partitioning a population into groups. The aim is to create segregated clusters wherein elements are similar to each other and dissimilar to elements of other clusters.

number of customers. On the other hand, as the average number of customers per route increases, there is a quadratic increase in computational time. As such, the sweep algorithm is extremely useful in instances where the total number of customers is large, but the number of customers per route is relatively small [96].

Petals algorithm

The sweep algorithm may naturally be extended to the petal algorithm which was developed by Balinski and Quandt [18] for smaller instances of the CVRP [124, 189]. The algorithm generates multiple routes, called petals, and selects an optimal set of routes by solving a set partitioning problem[3]. In this problem, the objective is to

$$\text{minimise} \quad \sum_{k \in S} c_k x_k \tag{3.1}$$

subject to

$$\sum_{k \in S} a_{ik} x_k = 1, \quad i = 1, \ldots, n, \tag{3.2}$$

$$x_k \in \{0, 1\}, \quad k \in \mathcal{S}, \tag{3.3}$$

where $\mathcal{S}$ denotes the set of routes, $x_k = 1$ if route k belongs to the selected solution and $x_k = 0$ otherwise. Furthermore, a_{ik} is a binary variable equal to 1 if customer i forms part of route k, and c_k is the cost of route k.

Fisher and Jaikumar

Fisher and Jaikumar [82] also proposed a heuristic with a cluster first, route second approach which, although similar to the sweep algorithm proposed by Gillet *et al.* [96], approaches clustering in a less trivial manner.

During the first phase, customers are assigned to one of K clusters (associated with K vehicles) based on the solution of a *General Assignment Problem* (GAP)[4]. The seed points of all clusters are determined by minimising the customer-to-seed distance, thus emphasising the importance of customer density [125, 200]. To simplify the GAP, the objective function is simplified by introducing a linear approximation to the cost of a route, based on the estimated insertion cost of a customer into a cluster. During the second phase, each cluster of customers is treated as a TSP, and a route is generated for each vehicle. More formally, the algorithm works as follows:

Step 1 Select seed customer $i_k \in N$ to initialise each cluster k.

Step 2 Calculate the approximated insertion cost d_{ik} of assigning customer i to cluster k as $d_{ik} = \min\left[c_{0i} + c_{ii_k} + c_{i_k0}, c_{0i_k} + c_{i_ki} + c_{i0}\right] - \left[c_{0i_k} + c_{i_k0}\right]$.

Step 3 Solve a GAP based on the associated d_{ik}-values, customer demands q_i, and vehicle capacity Q.

Step 4 Solve a TSP for each cluster of customers found in Step 3.

[3] A set partitioning problem aims to partition items in a given set into smaller subsets.

[4] In a GAP, the aim is to find the minimum-cost assignment of weighted items to knapsacks. Each item should be assigned to exactly one knapsack, without exceeding the capacity of any knapsacks [152].

This algorithmic approach is similar to the methods often followed by human route planners [125, 200]. When routes are planned based only on visual inspection, vehicles may be assigned to serve a particular neighbourhood or dense region of customers. Similarly, the clustering approach aims to create routes that are focused on high customer density.

A seed customer i_k may be selected based on automatic rules or by a human route-planner. The latter approach incorporates the knowledge of the subject-matter expert, appreciating that the experience of the human may contribute to the execution of the solution [82].

The manual selection of a seed point is investigated by Baker [15]. Initially, a route-planner is allowed to select a seed customer for each vehicle, after which the GAP is performed to assign customers to vehicles. This is further extended by enabling the route-planner to select two or more customers to outline each route, before performing the GAP assigning customers to the already outlined route based on an approximate insertion cost. This process increases the human influence, although it may be time consuming [16]. Baker and Sheasby [16] found that when applying the cluster first route second approach, the optimal solution of the first step does not guarantee the best overall solution.

Bramel and Simchi-Levi

Bramel and Simchi-Levi [35] introduced the location-based heuristic as a cluster first route second approach to solve the CVRP. In the location-based heuristic, the VRP is formulated as a location problem applied in telecommunications network design, namely the *capacitated concentrator location problem.* This problem is concerned with the placement of m concentrators to serve n terminals distributed across a Euclidean space. Each terminal has a demand q_i, for $i = 1, \ldots, n$ and are served by concentrators with a capacity of Q_j, for $j = 1, \ldots, m$. This approach is applied to the cluster first route second approach by seeing the concentrators as seed points for clusters. This heuristic therefore finds m seed locations and connects each to a feasible set of customers to serve. The aim is to minimise both the sum of the distances from the depot to the m seed points, as well as the sum of the distances between the n customers and their corresponding seed point.

Other applications of two-phased methods

Comert *et al.* [49] used a cluster first route second approach in a case study involving the delivery of goods from a depot to 78 retail stores. The performance of three different methods of capacitated clustering, namely k-means, k-medoids and random clustering, were compared. It was found that the capacitated k-medoids clustering algorithm outperformed the remaining clustering techniques on all metrics considered. Rautela *et al.* [165] also applied the cluster first route second heuristic to a real-world problem, aiming to minimise the cost of distribution for a cooperative dairy. The heuristic was implemented by employing capacitated k-means clustering in the first phase and the *cheapest length algorithm* in the routing phase. The end-delivery locations are clustered around intermediate DCs, after which the routing is performed for both the main depot to the DCs, as well as that of the DCs to their associated cluster. The delivery locations are therefore clustered based on their proximity, while adhering to the capacity constraints per DC. The approach produced high-quality results within reasonable computational times. Another approach, taken by Korayem *et al.* [123], involves the use of a relatively new biologically-inspired metaheuristic, namely the *grey wolf optimiser.* This metaheuristic is used in conjunction with k-means clustering to find feasible clusters during the first stage of implementation. Once k centroids have been determined, there are three methods of assigning customers to a cluster that are proposed. The first method involves merely assigning each customer to the nearest centroid, while the second method takes into account the capacity by incorporating a penalty cost when the capacity is violated. The third method assigns customers to a centroid, based on their distance from the centroid, their demand, as well as the violation in capacity caused.

3.1.3.3 Improvement heuristics

In improvement heuristics, existing feasible solutions are improved by performing exchanges within or across routes. This may either be applied to a single route or to multiple routes simultaneously. In the former, the improvement operates similar to heuristics aiming to improve TSP solutions. In the latter, multi-route structures are utilised and inter-route exchanges are made.

Single route improvement has generally been described in terms of the λ-opt mechanism, developed originally by Lin [129]. In this algorithm, a feasible route is improved upon by repeatedly exchanging λ links of the current route with λ different links. The exchange is only performed when the cost of the resulting route is less than that of the original route. In other words, the route is improved by removing λ links and reconstructing the route with λ different links in a shorter manner, possibly reversing one or more of the links. A route is defined to be λ-optimal if it is impossible to improve the route by replacing any λ links by any set of λ links [107].

As an example, a 3-opt move is illustrated in Figure 3.4. The original route is shown on the left, with three of its links, namely x_1, x_2 and x_3, selected to be removed. On the right, the severed links are replaced by new links, y_1, y_2 and y_3, in forming a shorter overall route.

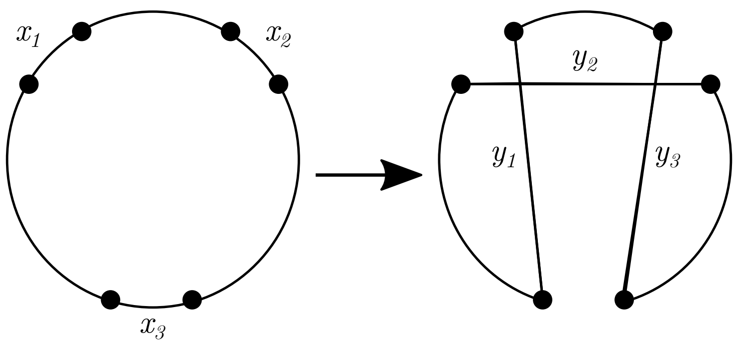


FIGURE 3.4: *An illustration of the 3-opt move adapted from Helsgaun [107].*

Note that the graphs shown in Figure 3.4 are not drawn to scale, and that the cost of the new links y_1, y_2 and y_3 are less than that of the original links x_1, x_2 and x_3. The 3-opt move is repeated until the route is considered 3-optimal, meaning no improving moves are available.

In an extension to the basic λ-opt algorithm, Lin and Kernighan [128] introduced an algorithm where the value of λ is varied. In what is now referred to as the *Lin-Kernighan algorithm*, the number of links that are to be replaced is varied at each iteration. Although this method is more involved and contain more complicated steps, it has been imperative for the optimisation of routes [116]. The improved methods that are based on this original λ-opt algorithm have allowed for the discovery of near-optimal solutions at a reasonable speed even for large instances of the TSP [116].

3.1.4 Metaheuristics

In the pursuit of higher quality solutions, the use of metaheuristics have become prevalent during the 1990s [124]. These algorithms perform an in-depth search of the solution space which often results in solutions that are near-optimal [54]. The increased prevalence of metaheuristics may be, in part, attributed to the growth of computational capacity over the years. A major distinction between the metaheuristic approaches and classical heuristics, is the ability of the objective function to deteriorate [125]. In other words, a metaheuristic approach may deteriorate from the local optimum in search of a solution tending even closer to optimality, whereas classical heuristics terminate when a local optimum is reached [189]. Although metaheuristics may lead to a more accurate solution, it is accompanied by the cost of longer computational times as well as a loss of simplicity [54].

Metaheuristics are primarily categorised as local search or population-based approaches [200]. Local search methods are focused on working with one solution at a time, escaping local optima, and avoiding cycling between solutions. Population-based metaheuristics approaches, on the other hand, are based on natural processes, utilising a population of solutions in pursuit of an optimal solution [191]. Specific variants of these two categories are further discussed.

3.1.4.1 Local search

Local search algorithms start with an initial solution x_0. At each iteration t, the current solution x_t is updated to x_{t+1} within a predefined neighbourhood $N(x_t)$ of x_t. This is true for $t \in [0, 1, 2, \ldots, T]$, where T denotes the maximum number of iterations allowed. The distinctions between different local search metaheuristics are typically governed by the definition of neighbourhoods and the traversal of solutions within these neighbourhoods [125]. Another aspect of consideration is the rules governing the termination of a local search algorithm. According to Toth and Vigo [189], three of the most common stopping criteria are:

- The objective function of the incumbent solution has not improved by a set percentage for a set number of iterations;
- The number of allowable moves in a neighbourhood has been less than a set percentage for a given number of iterations; and
- A given number of iterations have been performed.

Two popular local search metaheuristic algorithms, namely *tabu search* and *simulated annealing*, are discussed.

Tabu search algorithm

According to Cordeau *et al.* [54], the tabu search algorithm stands out as the best metaheuristic for application to the CVRP. This approach was first introduced by Glover [97] in 1986 and first applied to the CVRP by Willard [205] in 1989. By moving from the current solution x_t to the best non-tabu solution x_{t+1} in the neighbourhood $N(x_t)$, the local neighbourhood is explored. As the best non-tabu solution may be one that deteriorates the objective function, cycling between solutions is theoretically possible. To avoid cycling between solutions, revisitation of solutions that were recently examined are made *tabu* and are thus forbidden for a number of iterations [91].

Within the application to the CVRP by Willard [205], a neighbourhood is defined as all feasible solutions that may be attained when performing 2-opt and 3-opt exchanges in traversing solutions. This was improved upon by Pureza and França [161] while solutions are traversed by either moving a customer to a different route, or by exchanging two customers among routes while preserving feasibility. Neither of these algorithms, however, produced results that were adequate and further research into search mechanisms were required [91].

Osman [153] improved on the previous work by using the λ-interchange generation mechanism. Neighbourhoods are defined with a combination of exchanges, including 2-opt moves, reassignment of customers to different routes, as well as interchanges of customers between routes. In addition, two search strategies, namely *best admissible* and *first best admissible* were proposed and implemented. Both of the implementations produce excellent results, but still allow for improvement [91].

Simulated annealing algorithm

The simulated annealing algorithm was introduced by Kirkpatrick *et al.* [119] in 1983 as a widely applicable heuristic optimisation technique which produces results of good quality. They showed its power by successfully applying it to TSPs with several thousand customers. It was first applied to the CVRP by Robust *et al.* [169] in 1990, and later by Osman [153], producing promising results.

A key element of the simulated annealing algorithm is the introduction of stochasticity. At each iteration, a random solution is drawn from the neighbourhood of the current solution and evaluated for future use. Given that the aim is to minimise the function f and that the current solution at iteration t is x_t, the algorithm randomly selects a temporary solution x from the neighbourhood of the current solution $N(x_t)$. If this solution improves the objective function and $f(x) \leq f(x_t)$, it is selected as the next solution x_{t+1}. On the other hand, if the temporary solution deteriorates the objective function, it may still be selected by a given probability. More formally, if $f(x) \geq f(x_t)$, then

$$x_{t+1} = \begin{cases} x, & \text{with a probability of } p_t, \\ x_t, & \text{with a probability of } 1 - p_t. \end{cases}$$

The probability p_t is determined in each instance by a *temperature control* parameter, which changes over time based on a *cooling schedule*. The cooling schedule starts with high initial temperatures, allowing for a large exploration of solutions and avoids getting trapped in local optima. As the iterations progress, the temperature reduces, eventually not allowing any deterioration of the objective function [11].

Osman [153] adopted the simulated annealing algorithm, in combination with the Clarke and Wright savings method in solving the CVRP. An initial solution was generated using the savings method, which was followed by an implementation of the simulated annealing algorithm with an adaptive cooling schedule to improve the solution.

3.1.4.2 Population-based

Population-based approaches utilise a population of solutions in their search for the optimal solution [53]. Various natural processes, such as evolution, ant colonies, and swarm behaviour are imitated in an attempt to find interesting solutions. Two of these population-based algorithms, namely the *genetic algorithm* and the *ant colony optimisation algorithm* are discussed in more detail.

Genetic algorithm

Genetic algorithms draw inspiration from the natural evolution of a species. The algorithm is described in its entirety by the pioneer in this field, Holland [110]. As the general approach of the genetic algorithm requires little domain-specific information, it is widely applicable in a vast number of problems [189]. By imitation of the evolutionary processes of reproduction, mutation, and survival-of-the-fittest, a population of solutions are utilised to search the solution space.

A genetic algorithm is initiated with a number of candidate solutions named *chromosomes*, which make up the initial *population*. In an iterative fashion, the population will evolve by mechanisms emulating natural phenomena, namely reproduction and mutation [91]. In order to explain the basic workings of the generic algorithm, let $\mathcal{X}^1 = \left\{\boldsymbol{x}_1^1, \ldots, \boldsymbol{x}_n^1\right\}$ denote the randomly generated initial population of chromosomes. For each iteration $t = 1, \ldots, T$, apply Steps 1 to 3 k times with $k \leq N/2$, then apply Step 4.

Step 1 (Reproduction): Select two solutions from $\mathcal{X}^t$as parent chromosomes.

Step 2 (Recombination): Use the crossover operator to generate two offspring from the parent chromosomes.

Step 3 (Mutation): By some small probability, apply mutation to each offspring.

Step 4 (Replacement): Create a new generation of chromosomes $\mathcal{X}^{t+1}$ by replacing the $2k$ worst solutions from $\mathcal{X}^t$ with the $2k$ new offspring.

In the algorithm described, T denotes the maximum number of iterations, while k represents the number of selections per iteration. Once a stopping criterion is reached, the best solution produced up to that point will serve as the algorithm output.

Toth and Vigo [189] outline various applications of the genetic algorithm as applied to the CVRP. It is found that the use of this algorithm in VRPs is not yet competitive, especially when compared to local-search algorithms. This is as a result of the general applicability of the algorithm. When applied to a specific combinatorial problem, it has little capability to exploit information of the specific problem and, as such, cannot produce high-quality solutions in isolation [91]. When combined with local search methods, however, the resulting hybrid solutions may be designed specifically for the problem at hand [91]. Some of these hybridisations, such as employed by Prins [159], are found to be effective in finding solutions for the VRP [191].

Ant colony optimisation algorithm

The ant colony optimisation algorithm is based on the process of ant colonies foraging randomly for food, leaving traces to interesting food sources. Ants leave traces of pheromone on paths leading towards food sources, with the quantity of pheromone providing information to other ants. Colorni *et al.* [48] emulated this natural process in their proposed algorithm, with a colony of imaginary ants exploring the solution space. The objective function corresponds to the quality of food sources, and the adaptive memory mimics the pheromone trails [189].

The algorithm generally operates in two primary steps to find a solution [26]. First, candidate solutions are generated using the *pheromone model* which is a central component of the algorithm [26]. The pheromone model is a parametrised probability distribution over the solution space which is associated with the *pheromone values*. In the second step, the candidate solutions modify the pheromone values such that future sampling is biased toward high-quality solutions.

In applying the ant colony optimisation algorithm to the CVRP, it has been found that although good solutions may be produced on occasion, hybridisation is generally required [71]. Various hybrid models have been successfully implemented by using, for instance, Clark and Wright savings [166] or simulated annealing with 2-opt [36].

3.2 Computer simulation modelling

A model is an abstraction or simplified version of a complex, real-world system. The process of modelling is a methodology that is widely used to solve real-world problems. There are often cases where building or experimenting with real-world systems are expensive, dangerous or unrealistic. In such cases, an abstraction of the real-world system is preferred in order to perform experiments and analyses which may direct decision making in reality [28]. Models are merely simplified representations of reality, and may be in the form of physical models, mental models, or mathematical models. Mathematical models may be solved analytically or by means of simulation [126].

Simulation modelling may be defined as the imitation of the operation of a real-world process or system over time. It is used to describe and analyse system behaviour, evaluate future scenarios for the system, and contribute to the design or redesign of systems [20]. Simulation modelling may be applied to existing or conceptual systems, and serve as a planning and problem-solving tool in a broad range of applications.

Analytic solutions, as opposed to solutions generated by simulation, are wholly dependent on the set of input parameters provided to the model [29]. In instances where analytical solutions do not exist or prove difficult to find, simulation modelling provides an alternative methodology [126]. It provides a set of rules that governs the modelled system's behaviour and transitions between states over time. These rules may be in the form of equations, flowcharts, statecharts or other modelling constructs, while a collection of variables describe the state of the system at any given point in time [126].

3.2.1 Simulation model classification

In order to study a mathematical model by means of a simulation, it is important to define the different dimensions by which this may be achieved. A simulation model may be classified along three different dimensions, namely *static versus dynamic*, *stochastic versus deterministic* and *discrete versus continuous* [126].

3.2.1.1 Static-dynamic dimension

A static simulation model has no element of time involved. It is therefore either a model where time has no influence, or a representation of a model at a particular time. Examples of static models include Monte Carlo simulations and the simulation of multiple dice rolls. In these cases, the results only depend on the input parameters (which may include elements of stochasticity) and run independently of time [29]. Dynamic models, on the other hand, represent systems that change over time. A set of rules define how the model evolves over time, given its initial state [29].

3.2.1.2 Deterministic-stochastic dimension

A model that does not contain components of a random nature is termed a *deterministic* model. An example of such a model includes a system of differential equations modelling the nature of interactions between entities in a complex ecological system. In such a model, the future state is deterministic given the initial state and knowledge of the laws governing the interactions [126]. Although solving such models may be computationally intensive and time consuming, identical results will be achieved for each run. Once even a single component of randomness is introduced, a model is described as *stochastic*. The stochasticity incorporated into the model aims to reflect the uncertainty found in the real world. Stochastic behaviour can be incorporated either by modelling an input parameter as a probability distribution instead of a fixed value, or by incorporating chance into the behaviour of some element. The output of a stochastic model will inherently be stochastic in nature and should be interpreted as an estimate of the true nature of the model [126].

3.2.1.3 Continuous-discrete dimension

The continuous-discrete dimension refers to how the state of the modelled system changes during execution of a simulation run. In a discrete model, the state of the system changes instantaneously at discrete points in time. In a continuous model, on the other hand, the state of the system changes continuously over time. As an example, a system that is described by differential equations is continuously changing.

It may be difficult to decide between the use of a discrete or continuous model in simulating a real-world system. Depending on the level of analysis, a single real-world system may be modelled as either discrete or continuous. This decision is dependent on the objectives of the study [126]. Some models may entail a combination of discrete and continuous elements, in which the change of system state variables and global variables are different [20].

3.2.2 Steps in a simulation study

In conducting a simulation study, certain steps need to be completed in order to ensure thorough and sound practise. The correct and accurate construction of the model is dependent on the approach taken with respect to the study as a whole. A framework for performing a simulation study is proposed by Banks [20]. This framework is illustrated in Figure 3.5 and its constituent steps are detailed in this section.

1. *Problem formulation*: In this fundamental step, the problem statement should be formulated as clearly and entirely as possible. The simulation developer should take the utmost care to understand the client's description of the problem. This is often an iterative process wherein assumptions are clarified until agreed upon by both parties. As the project progresses, however, this stage may need to be revisited to account for new information or changes in client requirements.

2. *Project planning*: In this stage the objectives, scope and project plan are documented. The output of this phase is a detailed project proposal. It should include a statement of questions that need to be answered through the study and all the possible scenarios that need to be investigated. Finally, it should detail the technical elements of the project execution. This includes the timeline of the project, personnel requirements, as well as the planned use of hardware and software.

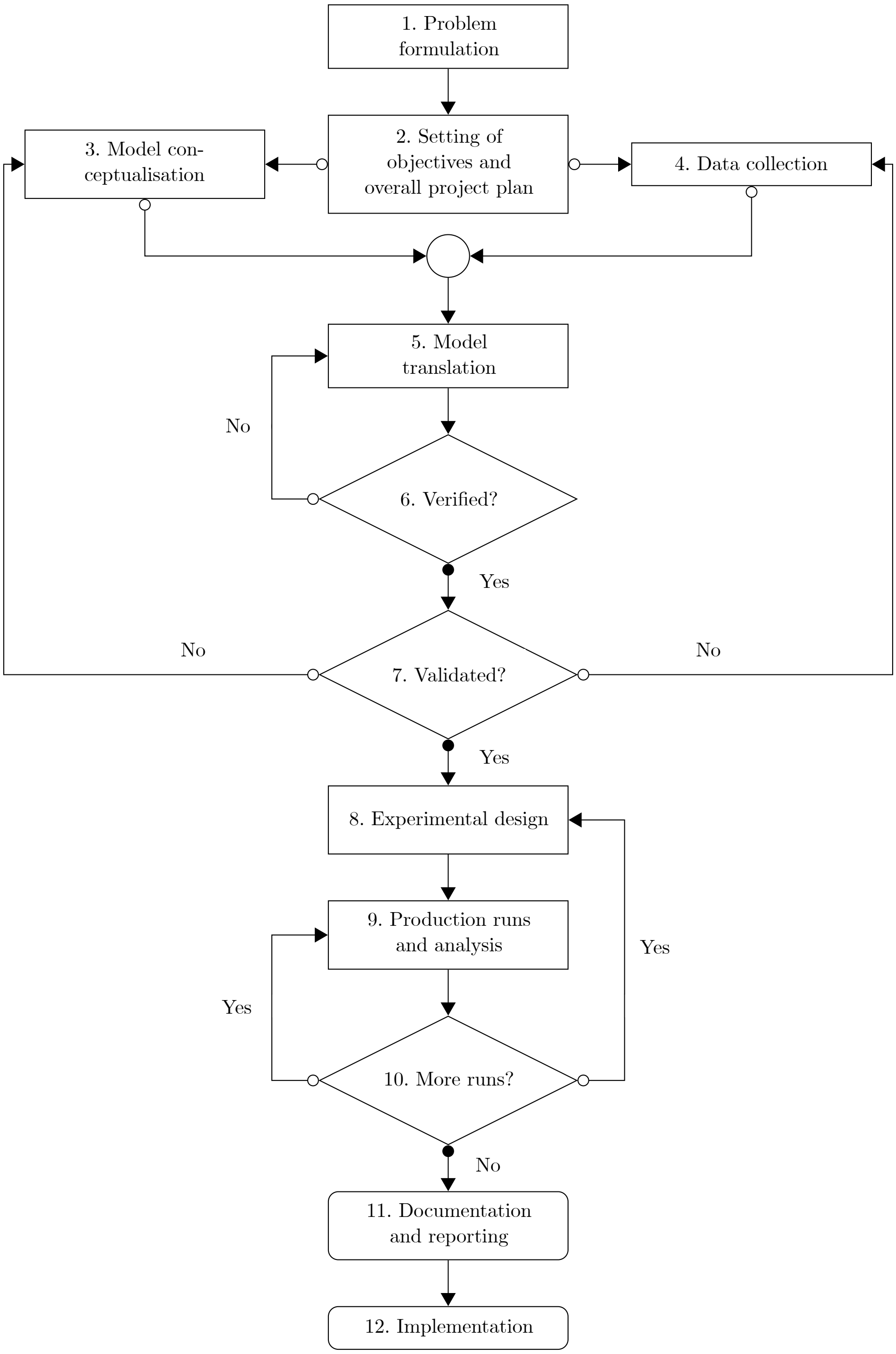


FIGURE 3.5: *Steps in a simulation study adapted from Banks [20].*

3. *Model conceptualisation*: During this stage, the real-world system is translated into a conceptual model. The components and structure of the real-world system are abstracted as logical and mathematical relationships. Thus, the basis of the model is established, with an initial focus on the most essential features. Once the essence of the real-world system is captured, complexity may be added iteratively, allowing the model to reflect reality more closely. The model conceptualisation should be performed in collaboration with the client, while considering the data availability, as well as the software capability. This will ensure that the model achieves the aforementioned objectives.

4. *Data collection*: In accordance with the garbage-in-garbage-out principle, the quality of the output produced by the model is limited by the quality of the input. The process of collecting data from the client and preparing it as input to the model is therefore an important phase. Since the data will impact the manner in which the model is constructed, the *data collection* and *model conceptualisation* stages should be executed concurrently. This will help ensure that the data required at each stage is readily available such that all objectives are achieved.

5. *Model translation*: In this phase, the conceptual model is translated into a computer simulation model. A programming language or simulation software that is appropriate for the selected modelling paradigm and project requirements is utilised to construct an executable model.

6. *Model verification*: Verification is the process of ensuring that the model is performing as expected. It is particularly concerned with the internal logic of the model and whether it is producing sensible results. Rather than performing the entire verification phase once the model has been fully developed, it should be executed as a continuous process during the model translation phase. This is especially crucial for large, complex models. Continuous debugging and prototyping forms part of this process.

7. *Model validation*: Model validation is the process of ensuring that the simulation model adequately represents the real-world system. It is achieved through comparing the output of the simulation model with that of the real-world system. If there are discrepancies that are greater than the acceptable level, the workings of the model should be investigated and improved upon through the use of data, case studies and expert opinions on the subject matter. Once the real-world system can be replaced by the model for experimentation purposes, validation is adequately performed. In the case where there exists no baseline for system comparison (*e.g.* in the case where the model represents a system which does not yet exist), additional methods of validation may be required. These include parameter variation and sensitivity analysis [171]. These methods involve varying the input values and internal parameters of a model and observing the associated effect on the model output. The effect of these inputs and parameters should be sensible and similar to that expected in a real-world system. Moreover, the identification of sensitive parameters (*i.e.* parameters for which a small change in input value has a large effect on model output) serve as an additional validation technique. These sensitive parameters should be adequately calibrated before model implementation.

8. *Experimental design*: In order to get results that are useful, the simulated scenarios, as set out in Step 2, need to be subject to experimentation. The length of simulation runs, the number of replications as well as the manner of initial conditions need to be determined. This is specifically important when stochasticity is introduced into the model, as multiple replications may be required to produce statistically significant results.

9. *Production run and analysis*: Model output is produced through the execution of production runs and the associated experiments. These results may then be analysed and used to answer the questions posed in Step 2.

10. *More runs*: Depending on the outcome of the initial production runs, additional runs may be required. The simulation analyst may decide that more runs of the current scenarios are required or that additional scenarios need to be examined. This may lead to the re-execution or redesign of current experiments.

11. *Documentation and reporting*: The two primary aspects that should be documented include the details of the simulation model as well as the results and analysis thereof. The simulation developer should aim to have a computerised model that is understandable, repeatable and reusable in case the project is extended or improved upon in the future. This is achieved through thorough documentation of the model details, assumptions and operating procedures. Furthermore, the documentation of results are core to answering the questions posed and meeting the objectives detailed at the beginning of the study. A clear picture of the model workings, the results captured and the interpretation thereof increases the client's confidence in making decisions based on the results.

12. *Implementation*: The implementation stage is the culmination of the successes of the previous stages. It involves implementing the suggested process or system configuration that was under investigation. The simulation analyst should, however, act as a reporter of the evidence and not as an advocate for the suggested system. Chances of success in this phase is further improved by constant communication and involvement from the client.

3.2.3 Advantages and drawbacks of simulation modelling

Simulation modelling has become an increasingly popular technique for analysing complex systems. Its popularity is justified by the benefits thereof which reaches into the fields of academia as well as business. There are, however, some drawbacks to simulation modelling as well and these should be considered. In creating a holistic picture of simulation modelling, the benefits and drawbacks of simulation modelling are detailed.

3.2.3.1 Advantages of simulation modelling

Simulation modelling holds advantages when compared to experimenting with the real-world system. First, the increased speed of simulated time allows for the study of complex phenomena, which may occur over longer timeframes in the real world, to be simulated in a fraction of the actual time [126, 134]. Furthermore, it allows for the evaluation of what-if scenarios which may be employed to experiment with various operating and environmental conditions. This may relate to the effects of decisions, such as policy changes, and significantly reduce the risk of experimenting with different policies in the real-world system. On the other hand, it may relate to variable environmental conditions which may be out of the decision maker's control. Simulation modelling also allows for experimenting with new or unknown systems where little data are available [134]. For such novel systems, simulation modelling may aid in testing the feasibility of the envisioned system and in evaluating alternative system designs before committing to the real-world investments [20].

Simulation modelling has additional advantages when compared to alternative methods of modelling. First, simulation allows for the study of stochastic real-world systems that are too complex to be evaluated analytically [126]. Furthermore, it encourages a systems-approach to problem solving, allowing for an overall view of the system, while being able to model the details within the system [134]. Furthermore, through various methods, such as parameter calibration and sensitivity analysis, deeper insight into the system may be generated. In particular, the sensitivity of the system output with respect to changes in different input parameters may be discovered. This often results in the discovery of bottlenecks or flaws within the real-world system [20, 134]. Finally, simulation packages often allow for advanced visualisation through real-time animation features. This guides both the verification and validation processes, and aids in communicating sophisticated models to non-experts [28].

3.2.3.2 Drawbacks of simulation modelling

Although simulation modelling serves as an excellent solution methodology for a wide variety of problems, it has its shortcomings. It is important to understand the limitations before venturing into a simulation study. The first drawback to consider is that the output of a simulation model with stochastic elements is inherently only an approximation of the model's true characteristics [126]. On the other hand, appropriate analytical models may easily produce the true model characteristics for a given set of input parameters. As such, even if several simulation runs are performed in approximating the true model characteristics, analytical models may be more applicable for the study of a given system [20]. Simulation models may, in some cases, be better suited for the comparison of different system designs rather than for optimisation [126]. Furthermore, conducting simulation studies are typically expensive and time-consuming. Often, when the high computational and monetary costs are avoided, the resulting model is not sufficient for the task at hand [20]. Finally, there may exist a mismatch between the confidence placed on the model results and the true validity of the model. The realistic animation and large amount of data generated in a simulation model often results in overconfidence in the output. The impressive nature of the model, even if the underlying reality is incorrectly modelled, may persuade one to trust the results [126].

3.2.4 Modelling paradigms

There exists a variety of approaches to use when imitating a real-world system by means of simulation modelling. It is possible to view a real-world system from various levels of abstraction ranging from a physical to a strategic level [28]. For each case, the aim of the simulation study may determine the level of abstraction that is appropriate or required. At the lowest degree of abstraction, physical-level models represent real-world objects with maximum detail. Focus is placed on physical interactions, dimensions and timings. For example, such a model would entail the physical parts of a vehicle or the behaviour of an individual customer. At the highest level of abstraction, on the other hand, individuality is lost and aggregated into an overall picture. Rather than considering individual vehicle parts or individual customer behaviour, the automotive industry or social and economic systems may be modelled [28].

Given the desired level of abstraction, there are various modelling paradigms which may be more or less applicable [29]. The adequate selection of a modelling paradigm contributes to the accuracy and realism with which the system is represented. According to Borshchev & Filippov [29], there exist three primary simulation modelling paradigms, namely *discrete event*, *system dynamics*, and *agent-based* simulation modelling. In Figure 3.6, the use of these modelling paradigms are captured, indicating at which level of abstraction each is applicable.

3.2.4.1 Discrete event simulation

Discrete event simulation is a paradigm wherein the state of the system changes at distinct moments during the simulated time [134]. Entities are objects with certain characteristics that are acted upon by the system and may represent people, parts, tasks and the like [29]. These passive entities move through the system and compete for limited resources, with queues managing their flow as they wait for occupied resources. When the entity reaches a resource and is acted upon, it is held by activity or delay time [20]. The commencement and completion of these activity and delay times trigger *events*. A mechanism allows for simulation time to jump forward to the occurrence of these distinct events, at which point entities are seized or released

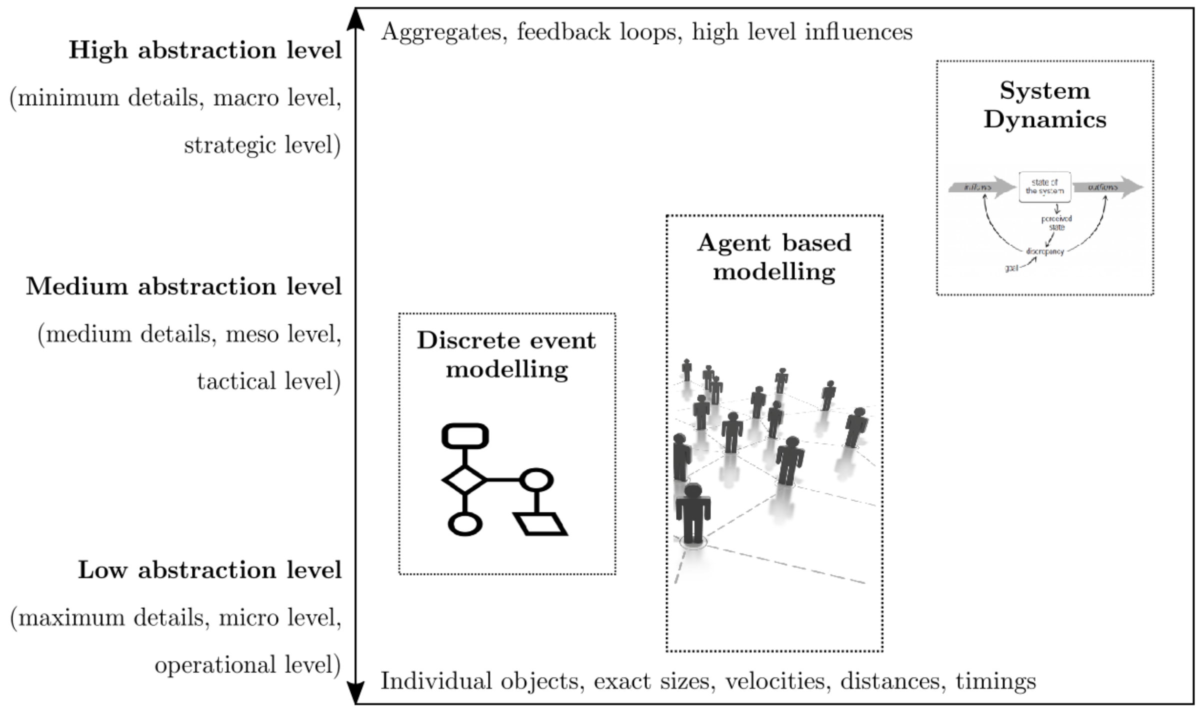


FIGURE 3.6: *Simulation modelling paradigms for different levels of abstraction adapted from Borshchev & Filippov [29].*

and instantaneous change may be brought about to both the entity characteristics and the state of the system. This modelling paradigm is generally applicable to models with a low to medium level of abstraction.

3.2.4.2 System dynamics simulation

The system dynamics approach follows a high level of abstraction and is mainly used for the modelling of long-term, strategic level instances, scenarios, or phenomena. It combines theory, methods, and philosophy to analyse systems in a wide range of fields including environmental change, politics, social and economic behaviour, and management [83].

The primary concept that system dynamic models utilise is that of feedback control [83]. Real-world processes are modelled as stocks (*e.g.* of material, people, money), flows between the stocks, and information determining the value of the flows [29]. The effect or influence of a stock on another may, in its simplest form, be modelled as a single reinforcing or balancing feedback loop by means of differential equations [29]. Multiple feedback loops are combined to form an interacting system of feedback loops and delay structures from which the system behaviour can be studied holistically.

3.2.4.3 Agent-based simulation

ABM is the computational study of agents interacting in an autonomous manner as evolving systems [115]. This paradigm follows a bottom-up approach and is ideal for investigating complex systems at a macro-scale by modelling individual agents within an environment. The global system behaviour is thus never explicitly defined, but emerges from the decentralised behaviour of individual agents [29].

O'Sullivan and Haklay [150] argue that agent-based models are analogous to an individualist view of society. In an individualist view, society is the direct result of aggregated individual behaviour. The notion of emergence in agent-based simulation models reflects this view, whereby complex phenomena of a simulated system often emerges from the collective behaviour of individual agents.

An agent may represent a variety of individual elements in a population, such as a household within a town, a biological cell within a more complex organism, or voters in a country. According to Macal and North [132], an entity is classified as an agent if it possesses a certain set of characteristics. An agent is a unique, discrete, and self-contained individual. It is clearly distinguishable from the environment within which it resides and the other agents with which it interacts. Furthermore, an agent has internal rules that govern its behaviour in the environment, its interaction with other agents and, ultimately, its goals that need to be achieved.

According to Bonabeau [27], there are three primary benefits to implementing ABM. First, *ABM captures emergent phenomena.* This follows from the notion that the whole is more than the sum of the individual parts as a result of the interaction between parts. The resulting global behaviour of a system that emerges is often counter-intuitive and would not necessarily have been captured when following a top-down modelling approach. Furthermore, *ABM provides a natural description of a system.* When modelling individual entities as opposed to overall system behaviour, the model may seem to reflect reality more closely. This makes ABM especially useful when the behaviour of individuals are complex, of a stochastic nature, or difficult to aggregate and treat as overall transition rates. Finally, *ABM is flexible.* The approach allows for flexibility in the size and scalability of models, the complexity of agents, and the levels of aggregation within the model. These aspects may be altered during the development process of the model which may ultimately improve the model's ability to adequately represent the real-world system.

3.2.5 Simulation model output analysis

Simulation models generally make use of random variables to emulate the stochastic processes of the real-world system being modelled. Consequently, the output of such a model may also be of a random nature and care must be taken in drawing conclusions from the output regarding the true nature of the system [126]. In order for inferences to be made about the true system from the output of a simulation model, it is necessary to make a number of assumptions about the underlying stochastic process. Statistical analysis of the simulation output may be impossible unless these assumptions are carefully considered [126].

A widely-used technique for interpreting the output of a simulation model include confidence intervals. Suppose that $x_1, x_2, \ldots, x_n$ denote the output of a certain variable x for n independent simulation replications. This set of observations may be considered as a sample of independent observations, drawn from the true population of the output variable. The aim of simulation output analysis is often to find an estimate of the true population mean μ and the population variance σ^2 [126]. Given a sufficient number of replications of the simulation run, resulting in a sufficient number of observations of x, a confidence interval may be constructed to provide a range of values within which μ will fall with a probability of $1 - \alpha$ where α is a values between 0 and 1. The confidence level is expressed as $100(1 - \alpha)$ percent.

In order to calculate such a confidence interval, a number of assumptions are required with respect to the sample of observations of x. In particular, it is assumed that the sample of observations is approximately normally distributed. From the *Central Limit Theorem*, it may be assumed that for a sufficiently large n, the sample mean $\bar{x}$ is approximately distributed as a normal random variable with a mean of μ, and a variance of $\frac{\sigma^2}{n}$ [45, 126]. Moreover, as population variance is often unknown in practice, it is often necessary to estimate the sample variance s^2 to be equal to the population variance σ^2. For a sufficiently large n, however, it is reasonable to assume that s^2 tends to σ^2. As such, given a sufficiently large n, the upper and lower bounds of a confidence interval for μ with a level of significance α may be approximated as

$$\bar{x} \pm z_\alpha \sqrt{\frac{s^2}{n}},$$

where z_α is the critical value derived from the standard normal curve, for an α level of significance [126].

The interpretation of a confidence interval for a selected α requires an understanding of coverage. Suppose an arbitrarily large number of independent confidence intervals are constructed, each utilising a sample containing n observations from the same population, it is assumed that $100(1-\alpha)$ percent of the constructed confidence intervals will contain or cover μ [126]. This proportion of confidence intervals that cover the mean is referred to as the coverage, and provides a degree of confidence in the output of a simulation model.

3.3 Microeconomic theory and consumer behaviour

> "Economics is the science which studies human behaviour as a relationship between ends and scarce means which have alternative uses" — Robbins [168].

The modern field of economics is diverse, covering a wide range of topics of which some may not appear, on face value, to be associated with the economy as it is generally understood. Economics, however, is not merely a field dedicated to studying only the economy, but also serves as a unique perspective through which the social world may be analysed [122]. Specifically, the economic perspective revolves around the concept of scarcity — the idea of having limited means with which to achieve ambitious ends. Under the conditions of scarcity, if resources are spent in one way it necessarily implies that it cannot be spent in another [122].

More formally, assume there is a set of alternatives A from which a single alternative a needs to be chosen by an individual. Furthermore assume that the alternatives may be ranked from $\{1, \ldots, n\}$, based on the expected satisfaction that would be experienced as a result of the selection. Once a decision is made to select a^1 for example, one naturally foregoes the satisfaction that would have been experienced through selecting any of the other alternatives. The forfeited satisfaction that is a consequence of making a decision under scare conditions is referred to as an opportunity cost. For example, the opportunity cost of choosing a^1 is equal to the satisfaction that might have been derived from selecting a^2, being the next best alternative.

The concept of opportunity cost allows for a better understanding of the decision-making process of individuals. For example, the portion of an individual's budget spent on luxuries cannot be used to save for the future; if an individual decides to study engineering, they may sacrifice the opportunity to study another degree at the same time; if time is spent commuting, the individual may forego the opportunity to spend the time on either leisure or productive work. To make intelligent and informed decisions, it is necessary to be aware of the value that is attached to the unrealised alternatives [122].

Microeconomics is a field of study where the general principles of economics are used to study the behaviour and decision making of individuals and firms [122]. Generally, when considering the decisions made by individuals, a completely rational consumer is assumed. Such a theoretically rational man, colloquially referred to as *Homo economicus*, was first considered by John Stuart Mill [141]. When an individual is said to behave as *Homo economicus* they are able to, while being aware of all alternatives, perceive all choice situations as a choice set. Moreover, they have a personal preference order over the alternatives, and will choose the best realisable alternative based on these preferences [122]. Additionally, it is assumed that "more is better," and that the rational man would always choose an alternative in which more can be consumed.

3.3.1 Utility theory and indifference curves

In accordance with the *Homo economicus* theory, when presented with a set of alternatives, an individual would aim to maximise the utility derived from the selected alternative. The utility of an alternative does not refer to objective usefulness *per se*, but is rather an indication of the perceived benefits as experienced by the individual [72].

More formally, consider the case where there exists a set of alternatives $A = \{a^1, a^2, \ldots, a^n\}$ from which an individual must choose. The utility derived from the alternative a is denoted by $U(a)$. Furthermore, consider the case where each alternative a represents a bundle of goods that may be consumed. This may be denoted as $a = (x_1, \ldots, x_m)$, where x_i denotes the quantity of good i available in the bundle. The consumer may have varying preferences for the available goods and, accordingly, varying preferences for the available bundles. In the case where there are two goods in a bundle, denoted by $a = (x_1, x_2)$, a consumer's preference may be illustrated by means of an indifference curve. In Figure 3.7, the indifference curve of an individual is illustrated when presented with varying amounts of goods 1 and 2.

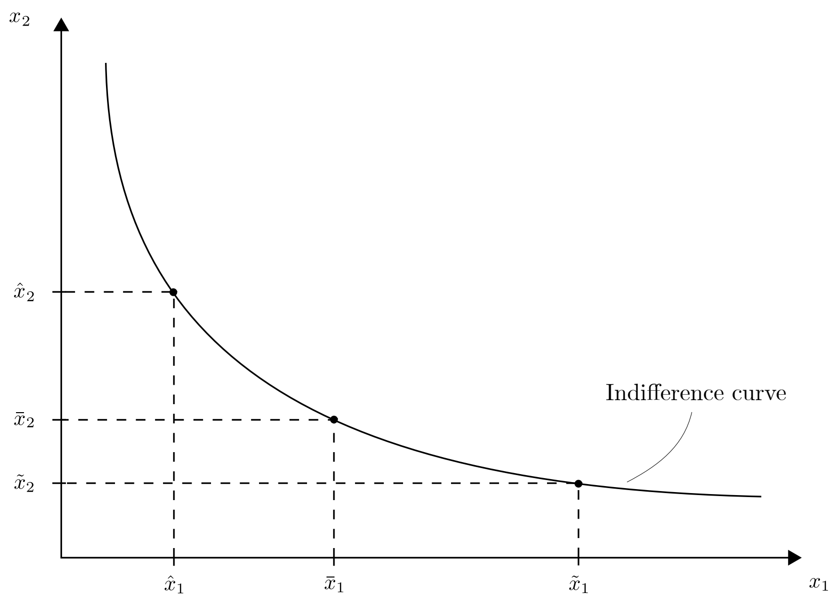


FIGURE 3.7: *The indifference curve of an individual for two goods.*

The downward-sloping curve represents the set of alternatives to which the individual is indifferent. To clarify this concept three unique bundles, $(\hat{x}_1, \hat{x}_2)$, $(\bar{x}_1, \bar{x}_2)$, and $(\tilde{x}_1, \tilde{x}_2)$, are indicated on the curve. For the individual in question, the utility derived from consuming these bundles are equal, such that $U(\hat{x}_1, \hat{x}_2) = U(\bar{x}_1, \bar{x}_2) = U(\tilde{x}_1, \tilde{x}_2)$.

In assuming that "more is better," it is implied that bundles containing more of both goods would result in a greater derived utility. As such there is a theoretically infinite number of indifference curves, each corresponding to a different derived utility. *Homo economicus* will therefore always prefer alternatives on an indifference curve associated with a relative higher consumption level. This concept is illustrated in Figure 3.8.

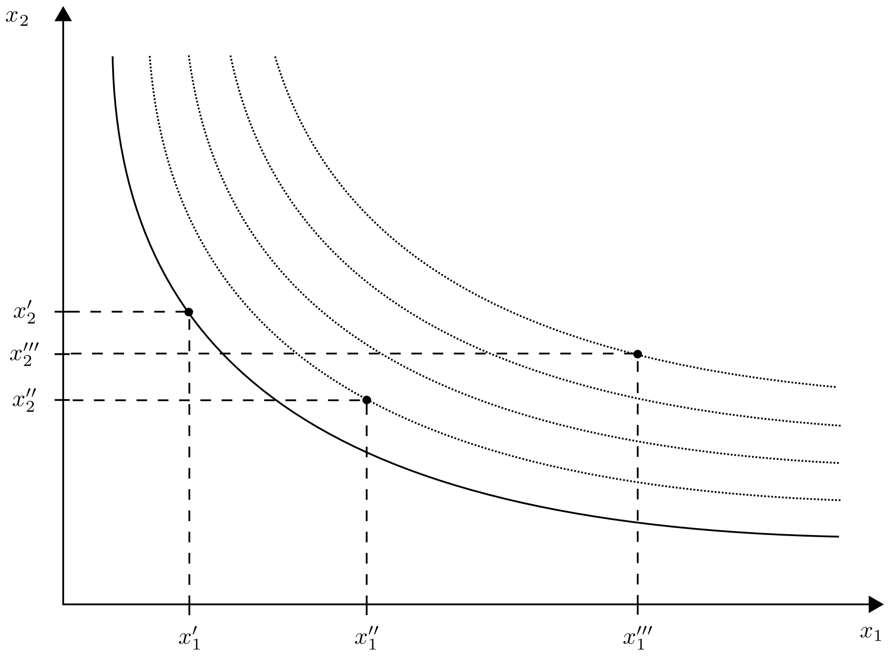


FIGURE 3.8: *An illustration of the infinite number of indifference curves for two goods.*

In a similar fashion to the previous example, this concept is clarified by considering three unique bundles, (x_1', x_2'), (x_1'', x_2''), and (x_1''', x_2'''), indicated in the figure, each on a different indifference curve. Each bundle is representative of a different derived utility, with higher indifference curves indicating higher consumption and, therefore, a higher derived utility such that $U(x_1', x_2') < U(x_1'', x_2'') < U(x_1''', x_2''')$.

Indifference curves may be utilised to understand the willingness of an individual to trade one good for another. The *marginal rate of substitution* (MRS) is the amount of one good a consumer is willing to give up in order to receive one more unit of another good, while maintaining the same level of derived utility. For the scenario depicted in Figure 3.7, the MRS represents the amount of x_1 the individual would be willing to give up to receive an additional unit of x_2. The MRS between good x_1 and good x_2 is equal to the slope of the indifference curve at any given point along the curve and is calculated as

$$\mathrm{MRS}_{x_1 x_2} = -\frac{dx_1}{dx_2}.$$

An additional calculation of the MRS between two good considers the individual's *marginal utility* of the respective goods. Marginal utility refers to the additional utility derived by consuming an additional unit of a particular good [72]. For a particular good x, the marginal utility is denoted as MU_x. Therefore, an individual's willingness to trade one good for another depends on their marginal utility for both goods. In particular, the MRS between good x_1 and good x_2 may be calculated as

$$\mathrm{MRS}_{x_1 x_2} = \frac{MU_{x_2}}{MU_{x_1}}.$$

The MRS between two goods is an economically meaningful concept, as it may provide insight into the preference ordering of an individual. Furthermore, this insight into the individual's preferences is irrespective of the utility representation used [122].

3.3.2 Perfect substitutes

A unique preference ordering exists where the indifference curve for an individual is a straight line. This implies that an individual is willing to substitute one good for another in a fixed ratio, regardless of the quantity of goods. In Figure 3.9, the indifference curve for the so-called *perfect substitutes* scenario is illustrated.

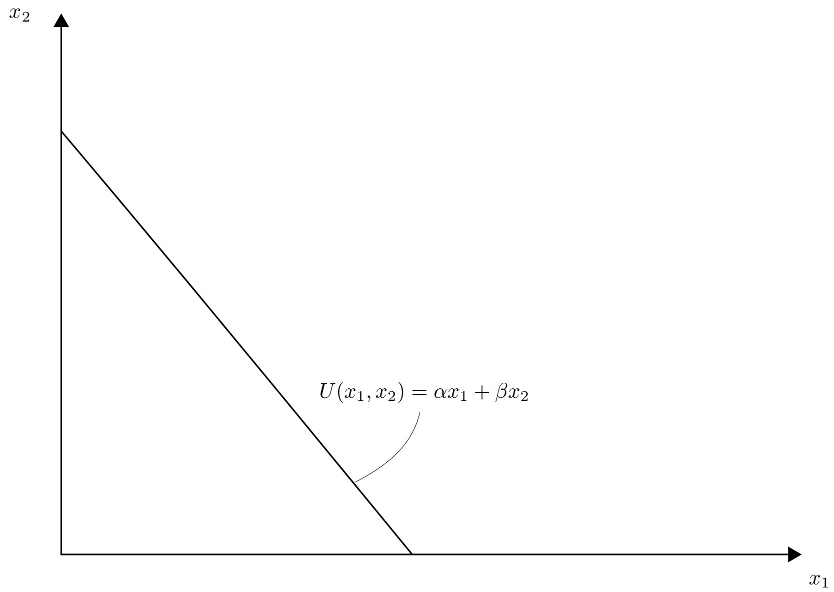


FIGURE 3.9: *The indifference curve for perfect substitutes.*

In the case where an individual's preference is such that x_1 and x_2 are seen as perfect substitutes, the utility derived may be calculated as

$$U(x_1, x_2) = \alpha x_1 + \beta x_2,$$

where α and β are constants. Furthermore, from Figure 3.9, it can be seen that the slope of the indifference curve for perfect substitutes is a constant. The MRS between goods x_1 and x_2 is calculated as

$$\text{MRS}_{x_1 x_2} = \frac{\alpha}{\beta}.$$

The constant $\frac{\alpha}{\beta}$ therefore captures the relative importance of good x_1 compared to good x_2.

3.4 The value of time

Transportation innovations, such as crowd logistics, are typically aimed at saving travel time, improving the reliability of travel time, or reducing travel costs [179]. These measures are often key drivers for individuals when making decisions related to personal transport alternatives. Small and Verhoef [180] found that the combined influence of travel time and reliability accounts for 45% of the perceived cost of a typical urban commuting trip. This overshadows vehicle operating costs, capital costs, and costs related to accidents.

Given the importance of time and time savings in transportation projects, a significant interest is placed on quantifying the value which individuals place on their time. The field of transport economics utilises various methods to estimate this value for travel time, generally through the use of discrete-choice models utilising utility theory. The interpretation to the value obtained (*i.e.* the MRS between travel time and travel cost) represents the individual's willingness to pay to reduce travel time by one unit [100].

According to Cesario [39], the interest in the value placed on travel time is two-fold. First, transportation projects are based on saving individuals' time. As such, the value that an individual associates with the time saving is of importance. Furthermore, there is a prevalent movement towards formulating models that take a disaggregated approach and consider behavioural elements. Travel time is often an important aspect in such models where individual entities are considered.

3.4.1 The trade-off between time and money

The theory of assigning value to time is based on the assumption that time is a resource that all individuals possess, and it may be allocated to different activities [100]. It is assumed that individuals perform this allocation in a manner that maximise their utility, given that the utility derived from their time depends on the activities that are being traded off [39]. Furthermore, it is assumed that the value derived from allocating time to an activity may be measured in monetary terms [100]. The generally accepted method for modelling such a subjective value of time relies on finding the MRS between travel time and travel cost. As such, the MRS between time and money is dependent on both the perceived scarcity of time and of money [114]. The value of time can therefore be seen as a traveller's willingness to pay for time savings under a particular circumstance [179]. It is noted by Armstrong *et al.* [10], however, that the value of time of an individual may not appropriately be considered as a single point value. Rather, the estimated value should be considered as a point estimate, reflective of an unknown probability function that underlies the individual's true value of time.

A distinction is made, however, between the decision to pay more for a shorter trip, or to accept a longer trip duration at a lower cost. The former is referred to as a traveller's *willingness to pay* (WTP), while the latter refers to a traveller's *willingness to accept* (WTA). The different measures of trade-off are illustrated in Figure 3.10. In this figure, the cost and time axes pass through the reference point r. The upper-left quadrant represents the WTP scenario where an individual must choose between the reference and a faster but more expensive alternative. The WTA scenario is depicted in the bottom-right quadrant, representing the choice between the reference and a longer but less expensive trip. Along with the distinction between these two scenarios, it is generally the case for any individual that WTP $<$ WTA. This general observation was made by Kahneman *et al.* [117] in 1990 and is referred to as the *endowment effect*.

With respect to transport innovations, the practical implications of the endowment effect may be considered as follows: An individual would be less likely to pay x amount of money in order to save t minutes, than they would be to endure t extra minutes travel time at a cost reduction of x amount of money.

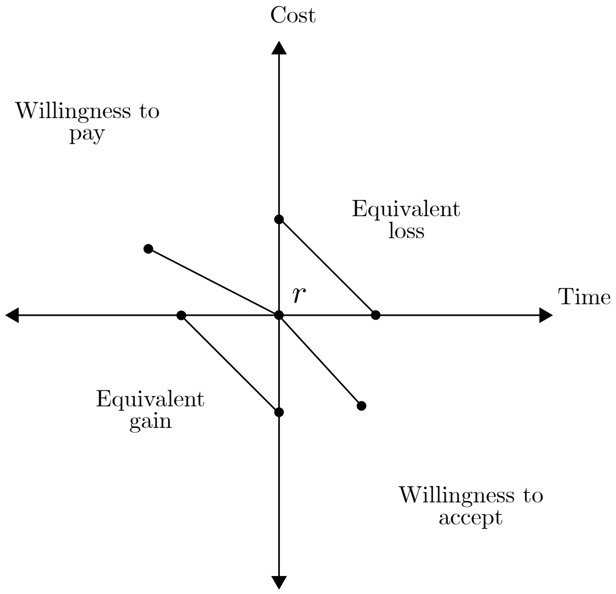


FIGURE 3.10: *The four quadrants defining the different measures of the trade-off between time and money adapted from De Borger and Fosgerau [67].*

3.4.2 Empirical studies

According to Small [179], the study of the value of time has three primary objectives. First, the value that commuters place on their time has implications for policy making and infrastructure developments within transportation. It may act as a useful tool within cost–benefit analyses of such investments [67]. Secondly, it elucidates certain aspects of human behaviour, which may be of interest in the broader field of economics. Finally, it is an important input to travel demand modelling. According to Gaudry *et al.* [89], valuing time in terms of monetary units is one of the best ways in which travel demand forecasting may be achieved.

In order to derive an estimation of the value that individuals place on their time and further determine the factors that influence this value, various techniques have been developed. Some of the most popular and widely-used techniques include stated preference models, revealed preference models, and models that estimate the value of time as a function of wage rate [176]. In stated preference models, surveys are utilised to inquire about an individual's travel preferences. Hypothetical scenarios relating to the trade-off between time and money are presented to an individual, followed by options of how they would act in such a situation. Their preferences are then derived from the answers provided, which allow for implicit equivalences to be subjectively established [89]. On the other hand, in revealed preference models, the behaviour of individuals are observed in reality and their preferences are derived from their actions [203].

In 1983, the United Kingdom Department of Transport examined the travel behaviour of commuters while selecting their mode of public transport with the implementation of both a stated and a revealed preference model [203]. The decision to travel either by bus or by train resulted in the trade-off between travel time and travel cost, with the bus being the slower, but less expensive option. The stated preference survey provided 16 hypothetical pairwise comparisons in which the commuter had to choose between travelling by bus or by train. Meanwhile, the actual behaviour and choices with respect to the selection of transport mode were observed in the revealed preference model. In general, the data collected in these models are used to determine commuters' willingness to trade their money for travel time savings.

Meta-analysis techniques may be used to combine results from various studies relating to the value of time. The aggregated results from such meta-analyses may then be used to predict the behaviour in situations other than which were explicitly studied [176]. Two prominent meta-analyses on the value of time savings were performed by Shires and de Jong [176], as well as by Abrantes and Wardman [1].

In the meta-analysis performed by Abrantes and Wardman [1], datasets were analysed with reference to variables that may explain the differences in how individuals value their time, specifically focusing on income disparities, travel purpose, mode of transport, and the type of study investigated. In the meta-analysis conducted by Shires and de Jong [176], a meta-model was applied to data from 25 European Union Member states. The records that were analysed are primarily representative of individuals with a middle to high income. The aim of the study was to produce new values of travel time savings and find the explanatory variables that cause variations.

Both meta-analyses found statistically significant results with respect to a number of explanatory variables. The primary variables that seem to influence the value of travel time savings include the country's gross domestic product (GDP) per capita, the travel purpose, the mode of transport, the distance of a trip, and the presence of congestion during the trip [1, 176].

3.4.3 Variables influencing the value of time

Trips may be classified according to the purpose of the trip. In estimating the value of time, distinctions are made between business trips, trips for commuting to work, and trips for other purposes such as leisure [1, 176]. It is seen that the value placed on the commuting time and all other non-business trips are very similar. Business travel time, however, seems to be valued significantly higher, reflecting the time-pressure often associated with business. Finally, the variations found across different leisure activities are deemed insignificant [176].

The different modes of transport to consider include air, rail, car, and bus trips. It can be seen that the value placed on air travel time is higher than that of car and rail [176]. Time spent on bus trips, in turn, are valued lower than that of car and rail time. The disparity in value of time across these transport modes may be explained by the difference in the group of people that tend to make use of each [176].

Shires and de Jong [176] found that of the variables considered in their study, the GDP per capita has the most significant impact on the value of time of an individual, with a positive relation exhibited. The effect is most notable in leisure travel, followed by commuting travel with a GDP per capita elasticity of 0.67 and 0.52, respectively. These results are substantiated by Abrantes and Wardman [1], where a GDP per capita elasticity of 0.899 was estimated — an even stronger effect. On the contrary, objections to these results are discussed by The International Transport Forum [114]. It is noted that there might be situations where an individual is "time poor" and "money poor" (*e.g.* a low income single parent), resulting in a higher value of time than would be deduced from their income. In such a case, the marginal utility of time influences the MRS significantly, resulting in a higher than expected MRS. As such, it may be important for studies to consider, in addition to the effect of income, the effect of lifestyle on the value of time [114].

Both Shires and de Jong [176], as well as Abrantes and Wardman [1], found the distance of a trip to have a significant influence on how the value of time is perceived. As trips increase in distance, the opportunity cost of the time spent becomes greater. Furthermore, the prospect of experiencing discomfort during long journeys influences the individual's perspective on time [1]. Thus, as the trip length increases, the perceived value per unit of time spent on the trip increases.

This effect is especially prevalent in leisure travel, as well as when using a car as mode of transport [1, 176].

A major difference with regards to the time value of money is seen in the value of peak *versus* off-peak travel, which is a result of the congestion associated with peak travel [1]. Congested travel time is found to be over 30% more valued than off-peak travel time. This phenomenon reflects the unpredictability and resulting anxiety and frustration of travelling during peak times.

3.4.4 Values from literature

For the purpose of the research conducted in this thesis, only a number of the influential variables are considered. Additionally, assumptions are made with respect to some of the influential variables, such as mode of transport, trip purpose, and trip length. Although crowd-shipping may be performed by other modes of transport, only car travel is considered in this study. Furthermore, with respect to the trip purpose, only non-business trips are considered. The values of time that are applicable to these overlapping categories are captured from the studies in question [1, 176] and converted to $/hour.

In Table 3.1, the value of travel time, as estimated by Abrantes and Wardman [1], are tabulated. Specifically, these values represent the monetary values of travel time for commute or other non-business travel purposes when using a car as transport mode. It is further segmented by the length of the trip.

TABLE 3.1: *Value of car travel time in $/hour for short to medium, non-business trips estimated by Abrantes and Wardman [1].*

Trip length	Commuting travel [per hour]	Other non-business travel [per hour]
Short	$6.60	$5.40
Medium	$8.40	$7.20

Similarly, in Table 3.2 the value of travel time savings, as estimated by Shires and de Jong [176], are summarised. Of the 25 countries investigated in this meta-analysis, the average value, minimum, and maximum values, are captured. Specifically, these values represent the monetary values of travel time savings for short distance commutes and other non-business travel purposes.

TABLE 3.2: *Summary of the value of travel time savings for short distance non-business trips estimated by Shires and de Jong [176].*

	Commuting travel [per hour]	Other non-business travel [per hour]
Average	$14.88	$12.48
Maximum	$22.06	$15.91
Minimum	$7.99	$7.00

3.5 Existing crowd logistics models

Crowd logistics is discussed in §2.5 with specific reference to a number of the real-world crowd logistics initiatives in §2.5.2. Mehmann *et al.* [138] suggest that research on crowd logistics is still in its infancy and that several ideas warrant further investigation. One of the identified pathways that could deliver deeper insights into the strengths of crowd logistics involves data analysis and simulation studies. As such, there has been a growing interest in investigating crowd logistics models using quantitative methodologies, such as mathematical modelling and optimisation [138].

In particular, mathematical and simulation models of companies utilising crowd-shipping for the improvement of their last-mile deliveries have become increasingly popular. Most notable are the works by Archetti *et al.* [9], Dahle *et al.* [63], Gdowska and Pedroso [90], Dahle *et al.* [64], and Dayarian and Savelsbergh [66]. In these models, a company may utilise both a dedicated fleet of delivery vehicles, as well as the services of crowd members to perform deliveries. The crowd, however, comprises in-store customers that may be willing to perform deliveries on their way home, referred to as ODs. A compensation is paid to the ODs for their efforts.

The models considered are categorised into three primary approaches, namely capacity-constrained, time-constrained, and agent-based approaches. The implementation and objectives of the various models vary to some extent and are discussed in the subsequent sections. The capacity-constrained approaches discussed include the works of Archetti *et al.* [9] and Gdowska *et al.* [90]. On the other hand, the time-constrained approaches include the models proposed by Dahle *et al.* [63], Arslan *et al.* [12], Dahle *et al.* [64], and Dayarian and Savelsbergh [66]. Finally, an agent-based approach proposed by Chen and Chankov [43] is discussed.

3.5.1 Capacity-constrained approaches

Archetti *et al.* [9] present a model to address the instance where in-store customers are utilised for performing crowd-shipping deliveries. They coin the term *VRP with Occasional Drivers* (VRPOD) in modelling the scenario as an extension of the static variant of the CVRP, extended to include a combination of regular deliveries and deliveries made by ODs. All online customer locations and demands, as well as the destinations of available ODs are known. The depot serves as the origin of all ODs, reflecting the idea that their willingness to perform a delivery is announced upon arrival at the store. Before travelling to their destination, ODs may be assigned any delivery for which the detour required is sufficiently small. If the deviation constraint is met, an OD will perform a single delivery and receive a small compensation in return.

The VRPOD is formulated as an integer program. It considers the set of customer locations, OD destinations, and the location of the depot. Furthermore, it assumes a sufficiently large number of dedicated capacitated delivery vehicles to fulfil the demand of online customers. The homogeneous fleet of delivery vehicles have similar capacities and incur the same costs and distances as they travel between the depot and customer locations. The model objective is to minimise the total cost of delivery, which comprises the cost of the regular delivery as well as the compensation paid to ODs.

Archetti *et al.* [9] assume that there is an infinite number of delivery vehicles available. This is based on the premise that, in practice, there should be enough vehicles to serve all customers. Delivery vehicles are assigned to routes that start and end at the depot, fulfilling the demand of multiple customers on these routes. Each vehicle, however, is capacitated and can only serve a fixed cumulative demand. The model may choose to not assign all customers to the dedicated fleet of delivery vehicles, and outsource certain customers to the crowd of ODs.

The crowd is defined to include only in-store customers that are willing to perform deliveries on an occasional basis. The set of ODs therefore has an origin at the depot, and known destinations that may be exploited by the model. An OD is able to serve at most one customer per trip, and it is assumed that the OD has the required capacity to meet the customer's demand. Furthermore, ODs are assumed to be readily available for delivery, although their willingness to deliver is dependent on a flexibility constraint.

A parameter ζ emulates the flexibility of an OD and governs their willingness to accept a proposed offer to deliver an order. Consider the scenario where an OD j is prompted to make a delivery to customer i, after visiting the depot h. The trip will only be accepted if $d_{hi}+d_{ij} \leq \zeta d_{hj}$, with $\zeta \geq 1$, where d_{hi} and d_{hj} denote the distance from the depot to the location of customer i and the destination of OD j, respectively, and d_{ij} denotes the distance between the location of customer i and the destination of OD j. This means, in essence, that an OD will accept a trip if the deviation made from its original route is less than $(\zeta - 1)$ times its original distance. The flexibility parameter of all customers are known and exploitable by the model, reflecting the assumption that ODs agree to how far they would be willing to detour upon registering.

Archetti *et al.* [9] propose two schemes to calculate the compensation paid to an OD for delivering an order. In both cases, the compensation is calculated after an OD had already accepted the offer and performed the associated delivery. The decision to deliver is therefore independent of the resulting compensation. In the first scheme, the OD is compensated with a fraction of the cost of delivering directly to the customer. When delivering to customer i, the compensation paid to OD j is calculated as ρc_{hi}, where c_{hi} denotes the cost of travelling from the depot to customer i and $0 < \rho < 1$. Note that the compensation is independent of the final destination of the OD, thereby only considering the location of the online customer. In the second scheme, OD j receives $\rho(c_{hi} + c_{ij} - c_{hj})$, where c_{hj} denotes the cost of travelling directly from the depot to the destination of OD j, c_{ij} denotes the cost of travelling from customer i to the destination of OD j, and $\rho \geq 1$. ODs are therefore compensated proportional to the deviation resulting from the delivery.

Archetti *et al.* [9] propose a multi-start heuristic, combining variable neighbourhood search and tabu search, in solving the VRPOD. In the first step, initial solutions are generated, assigning a number of customers to ODs, and the remainder to the dedicated delivery fleet. The solution space is then explored by swapping customers among routes.

The results of the computational study conducted indicate that substantial cost savings may be realised. This is, however, subject to having a large number of ODs with substantial flexibility available to perform deliveries. It is noted by Archetti *et al.* [9] that future research should consider the highly dynamic nature of the real-world setting. Specifically, the fact that ODs may arrive over time and are only available for a brief point in time. Furthermore, it is noted that the flexibility of an OD may be influenced by the compensation offered.

The model proposed by Gdowska *et al.* [90] is also a static representation of crowd-shipping. Similar to the model proposed by Archetti *et al.* [9], a set of online customers with known locations are served by a combination of a large, dedicated delivery fleet and a number of potential ODs. In an attempt to reflect real-world crowdsourcing more closely, however, Gdowska *et al.* [90] introduce a number of stochastic elements to the model. Most notably, it is not guaranteed that an offer to perform a delivery will be accept by an OD. If accepted, however, the company needs to pay a form of compensation. This is proposed in an attempt to model ODs as independent agents rather than dedicated employees. The objective of the model is to minimise the expected delivery costs, including the distance travelled by the dedicated delivery fleet and the compensation paid to ODs.

An algorithmic approach is followed to find the subset of customers that should be proposed to ODs, such that the expected delivery costs are minimised. The algorithm considers the location of customers to be visited, the cost of outsourcing a specific customer (*i.e.* the compensation paid to the OD), as well as the probability of an outsourced offer being accepted. To establish a baseline expected cost, the routing algorithm is performed for all customers (*i.e.* no customers are proposed to ODs). Thereafter, the expected delivery cost may be gradually decreased by proposing additional customers to ODs. Once a trip is accepted by an OD, the compensation fee must be paid by the system, even if it results in a deterioration of the objective function at a later stage.

Although Gdowska *et al.* [90] aimed to model ODs as independent agents, the model considers neither the number of ODs, nor their destination locations. Rather, each customer is assigned a random probability of being outsourced successfully. This approach emulates the probabilistic nature of a proposed trip being accepted by any OD. The probability, however, is both independent of the customer location as well as the characteristics of the OD population. Similarly, for a trip that had successfully been outsourced, the resulting compensation is determined randomly, independent of the location of the customer or the characteristics of the OD population.

This model is executed on a set of 15 customers that are served from a single depot. The locations of customers are randomly and uniformly distributed over a 1×1 Cartesian plane, while their demands are randomly sampled as an integer from $\{10, \ldots, 20\}$. A dedicated delivery fleet with a sufficiently large number of homogeneous vehicles, each with a capacity of 50 units, are employed and the cost of delivery is set equal to the total Euclidean distances travelled between customers. Both the probability of acceptance and the compensation offered are randomly and uniformly distributed in $\{0, 1\}$ for the set of customers, independent of their location.

The use of ODs resulted in an average cost reduction of 9%. It is furthermore noted by Gdowska *et al.* [90] that there is a need for further research into the calculation of adequate compensation values, as well as the probabilistic acceptance of proposed trips. It is noted that the optimum amount offered to an OD to serve a customer depends on the cost that would be incurred by the dedicated fleet for serving that customer. As such, compensation values should take into account the current dynamics of the last-mile delivery process. Additionally, the authors note that an OD's willingness to accept a trip should be based on their location (*i.e.* on the deviation made to serve a customer), as well as the amount offered as compensation. These elements, in addition to historical data on each individual ODs should be utilised in modelling probability functions that governs trip acceptance.

3.5.2 Time-constrained approaches

Dahle *et al.* [63] propose a dynamic and stochastic crowd-shipping model involving the probabilistic use of ODs that may complement the deliveries of a fleet of delivery vehicles. In this mixed integer model, however, a number of additional complexities are introduced. Most notably, a timeframe is introduced, comprising multiple decision-making epochs as well as a time window wherein deliveries need to be complete. At each decision-making epoch, routing decisions are made based only on information that has been revealed before that point in time. Furthermore, the arrival of ODs are modelled stochastically, with the information pertaining to their trips only revealed at a specific moment in time. The planning horizon is illustrated in Figure 3.11. The objective of the model is to minimise the cost of delivery, which includes the cost of the regular delivery vehicles, the compensation paid to ODs, and a penalty cost incurred when a customer is not served within the required time window.

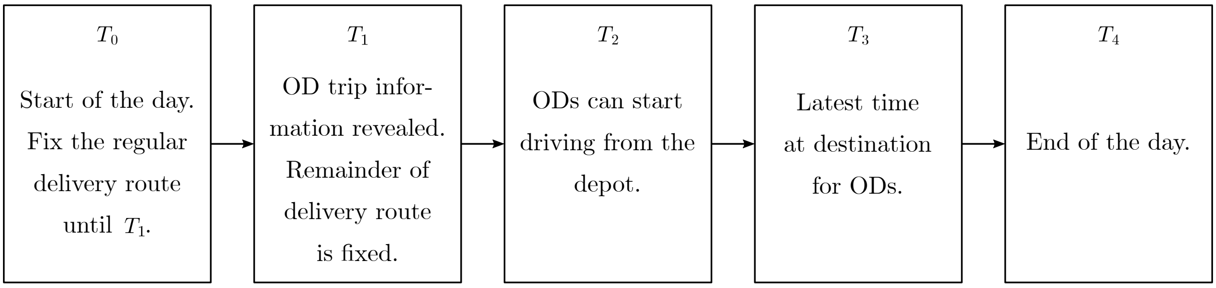


FIGURE 3.11: *The timeframe of the VRP with dynamic ODs adapted from Dahle et al. [63].*

At decision-making epoch T_0, the routes for the dedicated fleet of delivery vehicles are planned up to the next decision-making epoch, T_1. For this initial decision, the exploitable information include the locations of customers to be served, the costs and time associated with each route segment, as well as the destinations of the potential ODs. Although the OD destinations are known, it is uncertain whether or not they will be available at T_1. As such, the first phase of the delivery routes is planned with the possibility of ODs arriving at the depot later in the day. Moreover, routes may be planned optimistically (*i.e.* assuming all ODs will be available at T_1) or pessimistically (*i.e.* assuming no ODs will be available at T_1). An example of the first phase of routes is illustrated in Figure 3.12.

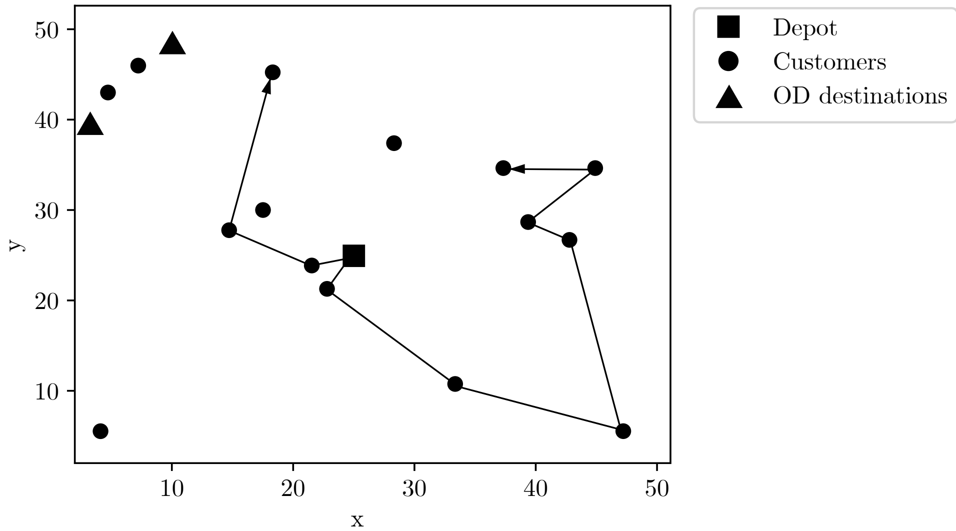
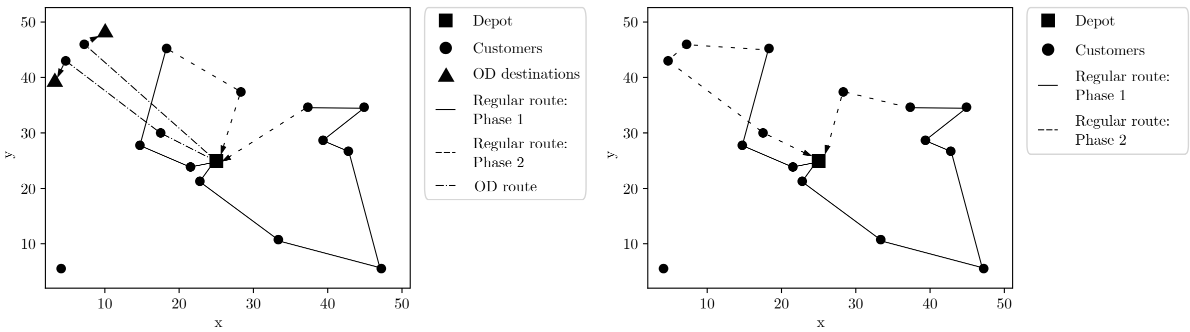


FIGURE 3.12: *An example of the planned delivery routes up to T_1 adapted from Dahle et al. [63].*

At T_1, more information is revealed to the system, at which point the remainder of the route is planned and optimised. This information specifically pertains to the availability of ODs at T_1. Given that an OD is available, they may be utilised from T_2 onwards to serve any number of customers. It is assumed that ODs are willing to perform any number of deliveries and have the required capacity to perform these deliveries. Furthermore, all planned trips should be such that all ODs may reach their destination by T_3. As compensation for their services, ODs are paid a price proportional to the cost of the extra distance travelled. In Figure 3.13, the routing solutions for the second phase of deliveries are shown for two different scenarios, illustrating the effect of the dynamic nature of the ODs. In the first scenario, illustrated in Figure 3.13(a), both the ODs arrived at T_1, whereas none of the ODs arrived in the second scenario, illustrated in Figure 3.13(b).



(a) *Routing utilising ODs at T_1* (b) *Routing with no OD arrivals at T_1*

FIGURE 3.13: *An example of the delivery routes planned from T_1, demonstrating the effect of the stochastic arrival of ODs adapted from Dahle et al. [63].*

In the case where both ODs arrived, one OD is assigned to serve two customers, while the second OD serves a single customer. In the case where no ODs arrived, the dedicated delivery vehicles are required to serve all customers. It can be observed that the first phase solution formed at T_0 considers the possibility of ODs arriving at T_1, as the planned routes end on a point where they would be able to adapt to both scenarios. Finally, it should be noted that, in both cases, the customer in the bottom-left corner is not served before the conclusion of deliveries at T_4. This is an indication that the cost of serving this customer within the required time window would be greater than the penalty incurred as a result of omitting the customer from the delivery route.

The model was executed with 20 delivery locations randomly distributed on a 50×50 square. Furthermore, two regular delivery vehicles are available, with either two or three ODs potentially arriving at T_1. The results obtained indicate that savings between 13% and 23% may be realised through the use of ODs.

In another implementation, Arslan *et al.* [12] consider a crowdsourcing platform that continuously receives delivery tasks and OD trip announcements. In this crowd-shipping model, a delivery task may originate at one of multiple pick-up locations for delivery to a customer within a promised lead time. Furthermore, ODs may announce their availability at any point in time. The model utilises an event-based rolling horizon, matching delivery tasks to available drivers at each decision-making epoch, while utilising only the information available to the system at that point in time. Any task that is not matched to an OD is performed by the so-called back-up delivery vehicles. The aim of the model is to minimise the system-wide delivery cost, which includes the compensation paid to ODs and the cost of utilising a back-up delivery vehicle.

Delivery tasks arrive over time and a delivery task p may be defined by a number of features. First, a limited number of pick-up locations exist in the model, representing multiple retail stores or depots. Each order originates at one of these locations at a certain point in time. This location and time is referred to as the task's pickup location, denoted by o_p, and earliest pickup time, denoted by e_p. Furthermore, the task has an drop-off location denoted by d_p and latest drop-off time, denoted by l_p. A system-wide promised maximum delivery lead time, denoted by L_p, governs the latest drop-off time of a package, such that $l_p = e_p + L_p$, ensuring that the promised service level is met. Delivery tasks arrive according to a uniform distribution over the defined model timeframe.

An OD may announce their planned trip k at any point in time during the model timeframe, while a number of defining features of the trip are captured. Once a trip is announced, the origin and destination locations of the OD, denoted by o_k and d_k, are made known to the system. Furthermore, for each trip an OD has an associated earliest departure time from the origin, as well as a latest arrival time at their destination, denoted by e_k and l_k, respectively. Finally, the OD specifies the maximum time that they would be willing to drive, as well as the maximum number of additional stops that they would be willing to make, denoted by T_k and Q_k, respectively. It is assumed that these restrictions (*i.e.* maximum time and number of stops) are more restrictive than the capacity constraints of the OD's vehicle. Similar to the delivery tasks, the OD trip announcements arrive according to a uniform distribution over the defined model timeframe.

Throughout the timeframe of the model, delivery tasks may be grouped together into jobs and matched to OD trips according to a number of constraints. The model utilises an event-based rolling horizon to determine feasible matches at each time t when a new task or trip announcement arrives. At each time t, the system considers all active delivery tasks and ODs, generating a set of feasible matches. For a feasible match, the total travel time required does not exceed T_k and the number of stops required does not exceed Q_k. Furthermore, the time schedules must allow for the parcel to be picked up after e_p and delivered before l_p, while the OD is allowed to reach their destination before l_k. Finally, precedence constraints ensure that for each task the package is picked up before it may be dropped off.

In addition to the ODs, backup drivers may be utilised to ensure that each task p is completed before its associated latest drop-off time l_p. Similar to the ODs, each backup vehicle b has an earliest departure time from the depot, denoted by e_b, and a latest arrival time at the depot, denoted by l_b. Thus, in addition to the matches made between OD trip announcements and jobs, matches are made for backup drivers and jobs at each decision-making epoch t. Such matches are feasible if they adhere to the schedule and precedence constraints.

The objective is to minimise the total delivery cost incurred. The cost of delivery is proportional to the distance travelled for both ODs and backup vehicles. Thus the system-wide cost per distance is constant. The matching problem is solved by means of an exact approach at each decision-making epoch.

Results are evaluated according to four KPIs, namely the total delivery cost, the fraction of tasks that were matched to ODs, the fraction of ODs that were matched to tasks, and the number of backup vehicles required. The study concluded that the use of ODs in addition to the backup vehicles are beneficial to the system, with reported cost reductions between 18.8% and 37%. It is further noted that this cost reduction is dependent on the distribution of pickup and drop-off locations.

Dahle *et al.* [64] propose a dynamic model that considers both time windows and vehicle capacities. In this model, a number of online customers with known locations and demands are served by a set of heterogeneous delivery vehicles in combination with a set of ODs with known destinations. The VRPPDTW is used as a basis of the model, extended to include ODs as an additional set of delivery vehicles. This formulation ensures that all deliveries are performed within the pre-defined time windows while adhering to the capacity and precedence constraints. The time windows of all orders, as well as the departure and arrival times of ODs, are known in advance of solving the model. The model objective is to reduce the total cost of deliveries, including the regular delivery vehicle costs and the costs of incentives paid to ODs.

One of the elements addressed by Dahle *et al.* [64], involves OD preferences and acceptance of offered trips. In order for a driver to accept a request, the compensation offered needs to be of an adequate amount. To model the effect of OD decision making on the final routing decisions, Dahle *et al.* [64] propose a three-step iterative procedure for outsourcing orders. In the first step, the company solves the extended VRPPDTW while utilising all possible ODs. Next, based on the currently proposed route and associated compensations, the ODs may accept or reject their assignments. This decision is dependent on both the compensation scheme utilised, as well as the individual compensation requirements of ODs. In particular, an OD k requires a minimum compensation of R_k^S to start a trip, and an additional factor R_k^C times the extra cost incurred as a result of the detour. Thus, when requested to perform assignment x incurring a detour cost of C_x for a compensation of $f(x)$, the OD k will only accept the request if

$$f(x) \geq R_k^S + R_k^C C_x.$$

In the final step of the iterative procedure, additional constraints are added to the VRPPDTW for each rejected request, prohibiting that particular assignment. This three-step process is repeated until all ODs have accepted the solution.

In the model proposed by Dahle *et al.* [64], the behaviour exhibited by ODs in the second step is known to the system, although it would be unknown to the company in reality. This allows the constraints of the final step to be added to the model pre-emptively, in effect removing the need for the iterative process. As such, the model is formulated to be solved once, providing the optimal solution while considering the threshold constraints of all ODs. In reality, however, these constraints are not known in advance. It is therefore impossible to add the constraints pre-emptively and impractical to add in an iterative process.

In the computational studies performed, the locations of online customers and the destinations of ODs are distributed randomly according to a uniform distribution across a square of size 50×50. Similarly, the start of customer time windows, as well as the departure and arrival times of ODs are randomly distributed across the planning horizon. The behaviour and capacity are assumed to be similar for all ODs. The computational results indicate that the use of ODs may result in cost savings in the range of 10–15%. Furthermore, it was shown that creating more complex compensation schemes that consider the behaviour of ODs more closely, may potentially lead to even larger savings. Finally, it is proposed that developing a model that is both stochastic and dynamic should be considered for future research.

In the study by Dayarian and Savelsbergh [66], a same-day delivery system with crowd-shipping is investigated. The system is abstracted as a dynamic and stochastic model. In-store customers register to participate in the innovative delivery solution beforehand, specifying the maximum amount of time they are willing to spend on a delivery. These participating customers notify the store when they are shopping and provide their post-shopping destination. Based on this information, the system may assign an OD any number of deliveries, given the resulting detour time remains lower than the maximum time specified by the OD. The placement of online orders and the arrival of in-store customers are stochastic in nature, and governed by a probabilistic arrival rate. A service time determines the maximum allowable time from the placement of an order to the delivery thereof. The use of in-store customers is supplemented by the company's fleet of delivery vehicles. The cost to the company for using an in-store customer is different to the cost associated with a dedicated delivery vehicle. Throughout the timeframe of the model, assignment and routing decisions are generated at each decision epoch. In particular, at every epoch the decision is made whether to deliver immediately to each online order, or delay the delivery in the hope that a more beneficial solution will be achieved during the next epoch. If the delivery is to be made immediately, the system furthermore decides whether to utilise a company vehicle or an OD for the delivery. These decisions are made with the primary objective of minimising the total lateness of deliveries with respect to the online customer's allowable delivery window, and the secondary objective of minimising the cost of delivery.

From the set of potential online customers, orders are received over time governed by a Poisson distribution with a set rate. For customer i, the time of placing an order is denoted by τ_i^0. Thereafter, the company aims to deliver within their guaranteed lead time, which is a denoted by L. A limited fleet of company vehicles are available to perform deliveries to online customers. Delivery trips made by company vehicles start at the depot, allows for deliveries to multiple customers, and terminates at the depot. The capacity of the vehicles are assumed to be infinite and the total cost associated with the company fleet is assumed to be proportional to the total travel time of deliveries. Although each order has a service guarantee, it is possible for the company vehicle to delivery an order late.

Similar to that of online customers, the arrival of ODs throughout the model timeframe is governed by a Poisson distribution. ODs arrive and announce their willingness to participate at a given rate. For an OD j, the time of announcement is denoted by τ_j^c. Once an OD has announced their willingness to participate, there is a waiting time that allows the OD to finish the process of driving to the store and/or of shopping. The parameters r and R defines the minimum and maximum waiting time, respectively, before an OD can commence with their delivery.

Each OD will, upon registering, define the maximum additional time they would be willing to spend on deliveries. This is captured in the parameter γ, and may be unique for each OD. With the use of this parameter, the willingness of an OD to make a delivery is defined within a coverage area. The assumption is made that an OD will be more likely to make a delivery on the periphery of their original intended route. As such, the coverage area is defined as the surface of an ellipse wherein a customer need not travel more than $(1 + \gamma)$ of its original trip time. More formally, for an OD with home location j, the region in which they are willing to deliver is defined by $t_{hv} + t_{vj} \leq (1 + \gamma)t_{hj}$, where v is a moving point along the border of the ellipse, t_{hv} denotes the time required to travel from the depot to v, t_{vj} denotes the time required to travel from v to location j, and t_{hj} denotes the time required to travel directly from the depot to location j. The region of acceptance is illustrated in Figure 3.14. For an online customer order i, with a latest dispatch time of θ_i, it can be assigned to an OD if the following conditions hold

$$t_{hv} + t_{vj} \leq (1 + \gamma)t_{hj}, \tag{3.4a}$$

$$\tau_j^0 \leq \tau_i^c + R, \tag{3.4b}$$

$$\tau_j^c + r \leq \theta_i. \tag{3.4c}$$

The equation in 3.4a governs the flexibility of the OD, only allowing deliveries to customers within the ellipse coverage area. The equation in 3.4b ensures that the online customer order is placed before the OD departs from the depot, while the equation in 3.4c ensures that the OD is finished with their shopping before the latest allowable dispatch time. These conditions therefore eliminate the risk of late deliveries.

The compensation of an OD comprises a fixed and variable cost. First, the fixed cost, denoted by c_f^c, is paid when at least one order is delivered by an OD. The variable cost, denoted by w, is proportional to the total extra time spent by the OD. Let the route travelled by OD i be denoted by Σ_i, and let the function $T(\Sigma_i)$ return the total time spent on the route. The compensation paid to OD i is given as $c_i^c = c_f^c + w(T(\Sigma_i) - t_{hi})$.

The model is developed to allow for both a static and a dynamic variant. In the static variant, the information pertaining to online order placement and OD arrivals are known in advance. Therefore, the assignment of customers to either ODs or company vehicles are performed in advance with full future information. The dynamic variant, on the other hand, advances over

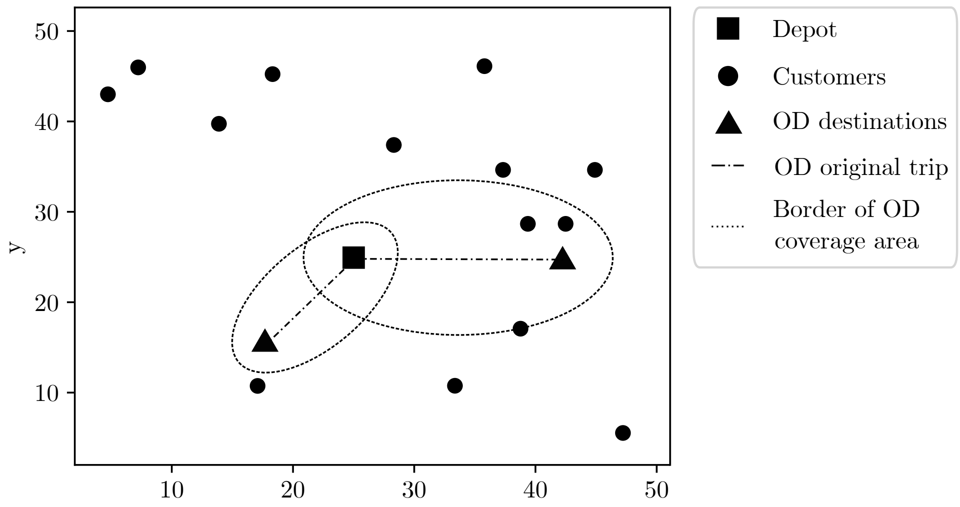


FIGURE 3.14: *The coverage area of ODs adapted from Dayarian and Savelsbergh [66].*

time with information being made available as time progresses. As information is made available and decisions are made, the state of the system subsequently changes. At each decision epoch, the state of the system is considered in deciding whether to assign an order to a company vehicle, to an OD, or to delay the delivery of the order. Each decision is made based on the time of the decision epoch, the orders that have been placed up until that point in time, the ODs that are currently available for deliveries, as well as the delivery vehicles that are available for dispatch.

The overall goal of the dynamic system is, therefore, to provide a set of actions at each decision epoch, based on the state of the system. The actions should provide the decision maker with an optimal set of delivery routes, assignment of customers to ODs and a set of customers whose delivery should be delayed. The overall actions should result in the minimisation of lateness and of total delivery costs.

In the computational instances generated by Dayarian and Savelsbergh [66], online customers and ODs are randomly distributed in a 100×100 square, with the depot located at the centre. All ODs are available for 30 minutes after announcing their trip, are able to deliver two orders, and has a flexibility parameter γ equal to 0.25. The compensation scheme utilised includes a fixed compensation of \$2.00 and a variable rate of \$0.50 per extra minute. The results indicate that convincing a larger set of ODs to participate in a crowd-shipping initiative may have a beneficial impact, both with respect to service quality and cost of delivery. Furthermore, it was found that by increasing the compensation offered to ODs, the percentage of ODs willing to participate increased, resulting in an improvement of service quality. It is noted, however, that the increased compensation often results in a higher overall delivery cost. As such, the authors note that the appropriate calculation of compensation to ODs is critical to the success of such a system. Finally, it is noted that the use of ODs are most beneficial when the last-mile delivery system is under stress. This includes cases when the delivery vehicle capacity is small, the service guarantee is strict, and visibility into future demand is limited.

3.5.3 Agent-based approaches

Chen and Chankov [43] propose an agent-based approach to model a system that relies completely on crowd-shipping for last-mile delivery. The model emulates a system in which multiple packages are available for shipping at one of multiple DCs. Each package has an intended end-destination (*i.e.* the location of an online customer) and an associated time window in which it should be delivered. Furthermore, ODs have pre-defined origin and destination locations that are independent of the last-mile delivery system. While travelling from their origin to their destination, an OD may choose to detour from their route to pick up one or more packages from a DC and deliver the packages to their associated destinations.

Three agent classes are proposed to represent and model the packages, DCs and ODs, within the system. A set of package agents become available daily, distributed among the various DC agents. OD agents, in turn, become active randomly throughout the day and travel from their origin to their destination locations. Once an OD agent is active, it identifies all packages that it could feasibly deliver while adhering to a number of constraints. In particular, the package should reach its destination within the required time window. Additionally, the detour resulting from delivering a package may not exceed the OD's maximum detour time. In the case where multiple packages may feasibly be delivered, the OD agent selects the package with the lowest detour time.

A number of KPIs are identified to assess the performance of the global system. These include the delivery service level (*i.e.* the percentage of packages delivered), the crowd utilisation (*i.e.* the percentage of ODs that delivered packages), the average number of packages delivered per OD, and the total detour time spent by ODs.

Chen and Chankov [43] indicate that the crowd-sourcing model is effective when economies-of-scale may be utilised. A large number of willing crowd members is required for a high service level. The crowd utilisation, however, decreases with an exceedingly high supply-to-demand ratio, as competition for packages emerges among ODs. On the other hand, for a low supply-to-demand ratio, a high incentive should be employed to encourage existing members to perform more deliveries on average.

3.5.4 Summary of existing models

The static VRPOD model proposed by Archetti *et al.* [9] serves as a good departure point for investigating the potential benefits and challenges associated with crowd-shipping. It was the first mathematical model to consider the use of in-store customers as potential parcel carriers. As an appraisal of the study, the authors note a number of limitations of the model. First, the model does not capture the dynamic and stochastic nature of real-world crowd-shipping. This includes the dynamic arrival rate of online orders and ODs alike, as well as the stochastic nature by which ODs accept proposed trips. Furthermore, as recommend by the authors, more sophisticated compensation schemes should be considered to include aspects such as the cost to serve the particular online customer being outsourced. In reflection, the inclusion of dynamic order and OD arrivals, an appropriate compensation scheme, as well as the behaviour of ODs require additional consideration. The existing models previously elaborated upon are briefly discussed with respect to these aspects.

The first aspect relates to the dynamic nature with which online orders arrive. In the models proposed by Archetti *et al.* [9], Dahle *et al.* [63], Gdowska *et al.* [90], and Dahle *et al.* [64], the online orders are deterministic and known in advance. As such, a single routing decision is made to satisfy the known demand while adhering to the various constraints. For the models proposed by Arslan *et al.* [12] and Dayarian and Savelsbergh [66], on the other hand, online orders arrive dynamically over the model timeframe. In the model by Arslan *et al.* [12], the arrival of orders are modelled as a uniform distribution for the duration of the timeframe, whereas Dayarian and Savelsbergh [66] model the arrival rate as a Poisson process. As such, multiple routing decisions are made at various points throughout the model timeframe. The dynamic approach reflects the real-world process of online orders more closely. Additionally, these models consider the scenario where decisions are made based on limited information about the future. As such, the associated dynamic models may be more useful in providing insight into the operation and decision making associated to a real-world crowd-shipping system.

In a similar fashion, the dynamic arrival of ODs at the depot is considered. In the approach by Archetti *et al.* [9], the full set of ODs is assumed to be available at the depot to perform deliveries. They note, however, that in reality ODs may arrive stochastically over time and may only be available for a short duration. Dahle *et al.* [63] attempt to address this issue by introducing the probabilistic arrival of ODs at a single point in time. This adaptation, however, only addresses the probabilistic nature of OD arrivals, while ignoring that ODs may appear at different points in time. Dahle *et al.* [64] capture the dynamic nature of OD arrivals by allowing for ODs to have different departure times. These times, however, are known in advance and may be exploited by the system and is therefore not an accurate reflection of reality. This is further improved in the models proposed by Arslan *et al.* [12] and Dayarian and Savelsbergh [66], where the arrival of ODs are modelled as dynamic and stochastic processes. In both cases, the information pertaining to OD arrivals are not known in advance and are governed by probability distributions. In particular, Arslan *et al.* [12] model the arrival of orders as a uniform distribution over time, while Dayarian and Savelsbergh [66] model the arrivals as a Poisson process. Similarly, in the model proposed by Chen and Chankov [43], ODs are modelled to arrive randomly throughout the model timeframe. Thus, in these models information is revealed as the model progresses, with the decisions made only exploiting the information available up to that point in time. This represent reality of a crowd-shipping system, including decision making under uncertainty.

Furthermore, the calculation of the compensation offered to ODs vary significantly among the different models. Archetti *et al.* [9] introduced two compensation schemes — one based merely on the online customer's distance from the depot, whereas the second scheme considers the deviation required by the OD to deliver the order. The latter compensation scheme is similarly implemented by Arslan *et al.* [12] and Dahle *et al.* [63]. A composite incentive scheme, composed of a fixed incentive and a variable rate per unit distance is utilised by Dahle *et al.* [64] as well as by Dayarian and Savelsbergh [66]. Finally, in the model proposed by Gdowska *et al.* [90], the compensation for a delivery is drawn randomly from a uniform distribution. It is noted that for effective crowd-shipping, compensation schemes should consider the cost-to-serve value of an order being outsourced [9, 90]. Finally, it is noted by Archetti *et al.* [9] that compensation schemes that depend on the OD's destination may have a number of challenges, particularly relating to the verification of the destination declared by the OD.

A mentioned in §2.4, a crowdsourcing task is generally presented to crowd members with an accompanying monetary incentive. Thus, the crowd member is able to evaluate the incentive and decide whether or not to perform the task. In the majority of the described models, however, the criteria by which an OD decides to deliver an order is independent of the compensation offered, which is typically calculated after-the-fact. In the model proposed by Dahle *et al.* [63], ODs have no autonomy and perform all deliveries assigned by the system. Gdowska *et al.* [90] attempt to introduce OD behaviour by modelling the acceptance of a proposed outsourced task as probabilistic. This is modelled as a uniform random probability, however, with no reference to the distance travelled by the OD or the location of the customer in question. In the models proposed by Archetti *et al.* [9], Dayarian and Savelsbergh [66], and Chen and Chankov [43], ODs define the maximum acceptable deviation that they are willing to make in advance. Throughout the execution of the model, ODs accept all orders for which the destination is within their predefined coverage area. Archetti *et al.* [9] consider the coverage area with respect to the distance an OD is willing to travel, whereas Dayarian and Savelsbergh [66] as well as Chen and Chankov [43] consider the additional time an OD is willing to spend. In the model proposed by Arslan *et al.* [12], ODs define in advance the longest additional time they are willing to travel, as well as the maximum number of deliveries they are willing to make per trip. In all of these cases, however, ODs have no autonomy beyond the initial declaration of their maximum deviation. Finally, the most sophisticated approach to modelling OD behaviour is proposed by Dahle *et al.* [64], wherein the acceptance of a trip is dependent on the compensation offered. A simplifying assumption, however, is made, dictating that the incentives required for an OD to be willing to deliver are known in advance to the system.

3.6 Chapter summary

The CVRP was introduced in §3.1, whereafter various solution methodologies were discussed. In particular, exact approaches, as well as heuristic and metaheuristic approaches were considered, noting their respective benefits and limitations. Simulation modelling was discussed in §3.2, with particular focus on ABM. This included a classification of simulation models, a discussion on the steps of a simulation study, the advantages and drawbacks associated with simulation modelling, as well as the various modelling paradigms. In §3.3, microeconomic theory and consumer behaviour was introduced, with particular reference to utility theory and indifference curves. This was extended into a discussion on the value of time and how consumers choose to trade their time for money in §3.4. Finally, existing crowd logistics models from the literature were discussed in §3.5.

CHAPTER 4

An agent-based model of customer crowd-shipping

Contents

The aim of this chapter is to describe the development of the proposed agent-based model of customer crowd-shipping. In §4.1, the modelling environment is described, including the AnyLogic simulation modelling environment, as well as the use of Python and the Pypeline custom library. A general description of the model is provided in §4.2. This includes descriptions of the real-world instance of crowd logistics that is considered and the data used as input to the model. Furthermore, the assumptions and limitations of the model are discussed. Thereafter, the model development is described in three phases. The first phase of the model development is outlined in §4.3, and pertains to the modelling of the last-mile delivery system of a traditional e-commerce retailer. This includes the generic formulation of a CVRP that is implemented and solved regularly during the execution of the simulation model. Furthermore, the heuristic methods utilised in solving the CVRP is described. The second phase, outlined in §4.4, pertains to the behaviour of ODs and their associated value of time. Finally, an integrated approach to customer crowd-shipping is developed in the third phase, which is described in §4.5. Penultimately, the KPIs of the model are described in §4.6. Finally, a description of the user interface developed for the model is provided in §4.7.

4.1 Modelling environment

In the following section, the modelling environment in which the model was developed is described. First, the AnyLogic software suite, which was utilised for the development of the agent-based model, is discussed. This is followed by a discussion on Python, which was used for a number of the calculations and visualisations, as well the PYPELINE custom library that allows the integration between ANYLOGIC and PYTHON.

4.1.1 The ANYLOGIC simulation modelling environment

The development of an agent-based model for the customer crowd-shipping initiative is conducted using the ANYLOGIC Personal Learning Edition 8.7 software suite. This software, developed by The ANYLOGIC Company, uses Java and object-oriented programming methods [29]. It allows modellers from various industries to translate conceptual models into executable simulation models in order to gain insight into complex systems and processes. ANYLOGIC is seen as a suitable choice, as it supports the agent-based approach and is flexible with respect to different levels of abstraction. Among its many useful tools, it allows for the real-time visualisation of the modelled real-world scenario through the use of animation. This makes interpretation and verification of the model more intuitive. The ease of defining agent behaviour, specifically by means of quantitative data, is extremely useful in a stochastic environment.

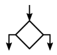
An agent-based model is developed by creating a population of agents and defining their attributes. This process is facilitated by various built-in functionalities such as statecharts, which govern agent behaviour based on their states. The relationship between agents, their interactions with the environment, as well as the transitions between states are defined by various parameters, variables, functions, distributions, and other modelling elements. In Table 4.1, a summary of the components used in the ANYLOGIC model is given.

TABLE 4.1: *The available components used in the* ANYLOGIC *environment.*

ANYLOGIC symbol	Feature	Description
	Agent class	Represents an agent class with some degree of autonomy that possesses attributes, behaviours, and relationships
	Parameter	Represents some static attribute or value relevant to an agent or to the environment being modelled
	Variable	Represents some dynamic attribute or value relevant to an agent or to the environment being modelled
	Function	Executes prescribed code or returns certain values when called in the model
	Event	Represents the activation of given actions within the model, triggered by a condition, rate, or fixed timeout
	Collection	Represents a collection of multiple similar entities, such as agents, grouped into a single unit
	Schedule	Returns a certain value that is dependent on a pre-determined time-based schedule
	Custom distribution	Represents a custom-made empirical distribution which returns given values at a certain probability
	Python Communicator	Library that allows for the execution of PYTHON scripts from within the ANYLOGIC simulation environment

In addition to the components summarised in Table 4.1, the ANYLOGIC environment makes use of statecharts to control agent behaviour. During the course of a simulation run, an agent may be in one of many states, represented by blocks. To transition between states, a number of built-in functionalities are available in the AnyLogic environment. The transition types and their associated symbols are described in Table 4.2.

Table 4.2: *The statechart transition symbols used in the* AnyLogic *environment.*

Statechart symbol	Transition type	Description
	Entry point	Points to the initial state of an agent
	Agent arrival	Represents a transition between states which occurs when an agent arrives at their destination
	Condition	Represents a transition between states which occurs when a certain condition is met
	Message	Represents a transition between states which occurs when an agent receives a particular message
	Time-out	Represents a transition between states which occurs when a set amount of time has run out
	Rate	Represents a transition between states at a rate governed by an exponential distribution
	Branch	Represents a transition where the destination state is determined by a condition

4.1.2 Python and the Pypeline custom library

Python is an open-source programming language that is well-equipped for integrating software systems [198]. Some of the benefits of Python include its open-source nature, its vast documentation, and its ability to integrate with a large number of community-contributed libraries. It is widely used for scientific and numeric computing and integrates with a number of packages to enable the solving of combinatorial optimisation problems. Furthermore, Python libraries such as matplotlib allows for data visualisations with a high degree of customisation.

To exploit the benefits of Python in developing an agent-based model, the Anylogic simulation software is integrated with Python. While the simulation model is built and run in AnyLogic, the complex computations and data analysis is outsourced to a number of Python scripts. The Pypeline custom library by Wolfe-Adam [207] allows for this integration of Python and Anylogic. The modeller is able to write Python scripts which may be called during model execution. The library is imported into AnyLogic and accessed through the use of the PyCommunicator object. The use of this object allows some of the AnyLogic components to be translated into a `.json` object, which may be read into a Python script as a dictionary. Most notably, agent populations or subsets of agent populations may be translated into a Python dictionary and their attributes used in calculations.

4.2 General description of the proposed model

Customer crowd-shipping involves the last-mile deliveries of an omni channel retailer, serving both online and in-store customers. Registered online customers place orders throughout the day, while in-store customers may shop by visiting the brick-and-mortar store. To fulfil the orders placed by online customers, the retailer has a dedicated fleet of delivery vehicles that perform deliveries at scheduled times throughout the day. In addition to these regular deliveries, the company may outsource some of the orders to a number of willing in-store customers, referred to as ODs.

The customer crowd-shipping initiative is modelled in accordance with the principles of crowd-sourcing business models as discussed in §2.4. The retailer (*i.e.* the crowdsourcer) aims to outsource delivery tasks to a subset of their in-store customers that have registered as ODs (*i.e.* the crowd). This involves the provision of an open call by the retailer, by means of a mobile application or website (*i.e.* the crowdsourcing platform), followed by the intentional selection of a task by an OD (*i.e.* a particular crowd member). ODs are modelled as independent agents that act in their own self-interest, rather than as an extension of the retailer's dedicated fleet of delivery vehicles. Furthermore, the open call by the retailer merely involves presenting potential ODs with a list of online orders to be delivered, accompanied by the associated incentives offered for the delivery of each order. The decision to deliver any of the proposed orders, however, is ultimately made by an individual OD, aiming to maximise their perceived personal utility. In particular, it is modelled such that an OD will compare the incentive offered to the value associated with the additional time required to perform the delivery.

The retailer in question therefore has limited control over the ultimate delivery routes executed. A degree of control, however, is possible by selecting which orders should be proposed for crowd-shipping, the nature of the compensation scheme, as well as the decisions relating to the deliveries performed by the dedicated fleet. Although modelling ODs as crowd members with autonomy may result in routes that are sub-optimal, the reality of a crowdsourcing initiative is reflected more adequately.

The model is developed and described by means of three phases. In Phase I, the traditional operations of an e-commerce retailer with last-mile deliveries are modelled. This includes orders placed by online customers being fulfilled by the dedicated delivery fleet. Phase II of the model development involves the inclusion of ODs, focussing on modelling their behaviour, as well as their perceived value of time. The final phase of the model development builds on the processes developed in Phase I and II. The behaviour of ODs is integrated into the traditional retailer operations, resulting in a last-mile delivery system utilising crowd-shipping in addition to the dedicated delivery fleet. In Phase III, the details relating to this integrated approach are described with respect to the proposed cost-to-serve algorithm, the proposed incentive scheme, the selection of orders for crowd-shipping, and ultimately, the decision-making process of ODs. Access to the agent-based model and it associated components is provided by means of a GitHub repository, as detailed in Appendix B.

The agent-based model, as developed in the ANYLOGIC software, creates an object class, namely `main`, wherein the environment of the model is established. All agent classes reside, move, and interact within a continuous two-dimensional simulated space contained within the `main` object class. Each distance unit in simulated space represents 100 meters in real space. In addition to the continuous space, a simulation run is executed over a simulated time dimension, with a minute as time unit (*i.e.* one second represents a minute of simulated time).

A number of object classes are established within the `main` object class. First, the single retail store is captured by the `depot` agent class. Furthermore, online customers are modelled as the `customer` agent population. The `delivery_vehicle` agent class models dedicated delivery vehicles that perform the traditional last-mile deliveries. Finally, the `OD` agent population models in-store customers that are registered as ODs.

4.2.1 Model data

As customer crowd-shipping is a modern innovation still in its infancy in practice, no real-world data is available for the modelling thereof [9, 90]. As such, this model makes use of theoretical data. Similar to Archetti *et al.* [9], this model exploits the Solomon instances of the CVRP, which are well known and widely available [183]. These problem sets provide a routing and scheduling environment and are commonly used as benchmark input data for solving the VRPTW. Additional information on the problem sets used are provided in Appendix A. As time windows are not considered in this model, these problem sets are used to define the location of a depot (x_h, y_h), the locations of the online customers (x_i, y_i), the associated demands of all customers, as well as the capacity of a homogeneous fleet of delivery vehicles. The coordinates of the online customers in the dataset fall within a 100×100 space, which represents a real space of 10×10 kilometre.

The location of the ODs, on the other hand, are distributed uniform randomly in the square modelled space with lower left hand corner $(\min_i x_i, \min_i y_i)$ and upper right hand corner $(\max_i x_i, \max_i y_i)$. The locations of the customer agents, OD agents, and the depot agent are implemented at the initialisation of a simulation run.

4.2.2 Model assumptions and limitations

A number of assumptions are required to establish the proposed agent-based simulation model for customer crowd-shipping. Although these assumptions may simplify the model, it is still able to offer a fair representation of customer crowd-shipping. The assumptions may be broadly categorised as assumptions relating to the VRP, the online customer behaviour, and the behaviour of ODs.

4.2.2.1 Assumptions relating to the CVRP and delivery vehicles

The first set of assumptions considers the CVRP and the associated delivery vehicles of the retailer.

1. *Euclidean distances are utilised.* The distances considered in the model are calculated as the Euclidean distance between customer agents, OD agents, and the depot agent.

2. *Delivery vehicles are homogeneous.* The fleet of dedicated delivery vehicles are homogeneous with respect to their capacity, the constant speed at which they travel, as well as their operating costs.

3. *Operating costs are proportional to the distance travelled.* The fixed cost of operating a vehicle (*e.g.* the maintenance and wages) are therefore assumed to be included in the distance-rated cost of delivery.

4. *There is an abundance of delivery vehicles available.* The dedicated fleet of delivery vehicles are capable of serving any number of customers regardless of their demand.

5. *Time windows are not considered.* Although the model is dynamic, there are no time windows involved for the CVRP. As such, all customers are served for each instance of the CVRP.

4.2.2.2 Assumptions relating to online customer behaviour

Furthermore, a number of assumptions pertaining to the behaviour of online customers in the system are made.

6. *No online customers register or deregister during the simulation period.* The set of online customers considered remains constant for the duration of the simulation model execution. This implies that no new online customers may register and that no existing online customers are lost during a simulation run.
7. *An online customer cannot place an additional order while waiting for a delivery.* Only once their initial order is fulfilled may they place an additional order.
8. *There are no failed deliveries to online customers.* This implies that when either a delivery vehicle or an OD arrives at an online customer's home location, the delivery would be successful.

4.2.2.3 Assumptions relating to OD behaviour

Finally, a number of assumptions are made with respect to the behaviour of ODs.

9. *No ODs register or deregister during the simulation period.* The set of ODs considered remains constant for the duration of the simulation model execution. This implies that no new ODs may register and no existing ODs are lost to the system during the simulated period.
10. *ODs travel to their home location after visiting the depot.* When considering the deviation made by an OD, it is assumed that the OD travels to their home location after delivering an order. It is not, however, necessary to assume that the OD starts their journey to the depot from their home location.
11. *ODs act as Homo economicus (or rationally).* This implies that when considering their alternative options, ODs are aware of all alternatives, they are able to rank the alternatives based on their preferences, and they will select the alternative that maximises their perceived utility.
12. *ODs are capable of serving only one customer per trip.* It is assumed that ODs travel with their personal vehicle, as opposed to the delivery vehicles used by the retailer. The assumption is made, however, that an OD has the capacity to serve a single customer, regardless of the demand of the customer.
13. *An OD considers the time spent travelling as more valuable than the physical cost of travelling.* As such, their willingness to perform a delivery is limited by the additional time required, rather than by the cost associated with performing the delivery.

4.2.3 Simulation timeframe

To model the working hours of a traditional retailer, a schedule `depot_open` is utilised. This schedule acts as a binary variable that is set to TRUE during working hours and FALSE during hours in which the depot is closed. Although the model user may adjust these values to the hours fitting their unique retailer, it is assumed in this case that the retailer remains open between the hours of 08:00 and 17:00.

4.3 Phase I — Online shopping and last-mile delivery

In the first phase of the model development, the traditional online shopping system of a retailer is replicated in the model environment. This includes the placement of orders by online customers and the fulfilment of order deliveries to these customers. For this phase of the model, the deliveries are only performed by the dedicated fleet of delivery vehicles and ODs are not considered. The agent classes involved in Phase I are therefore the `depot` agent, the `customer` agent population, and the `delivery_vehicle` agent population.

The agent populations are introduced into the model upon initialisation of a simulation run, and a number of their associated parameters are defined from the selected dataset. Each agent from the `customer` agent population, as well as the `depot` agent, is spatially placed in the modelling environment according to their Cartesian coordinates which are captured in the agent's `coordinates` parameter. Furthermore, the demand associated with each customer agent is captured in the `demand` parameter. Finally, agents of the `delivery_vehicle` agent population are placed at the depot, with their capacity captured in the `vehicle_capacity` parameter stored within the `main` agent environment.

4.3.1 Online customers and order placement

Modern-day e-commerce websites require users to register in advance of placing orders online to ensure that the necessary data relating to the consumer are captured. Companies therefore typically have a database with the information of all registered customers. Similarly, the `customer` agent population represents the entire set of potential online customers for the given retail store. As such, a customer agent initially exists in a passive state until they place an order, at which point they become active in the system. The Phase I statechart of the `customer` agent population is shown in Figure 4.1.

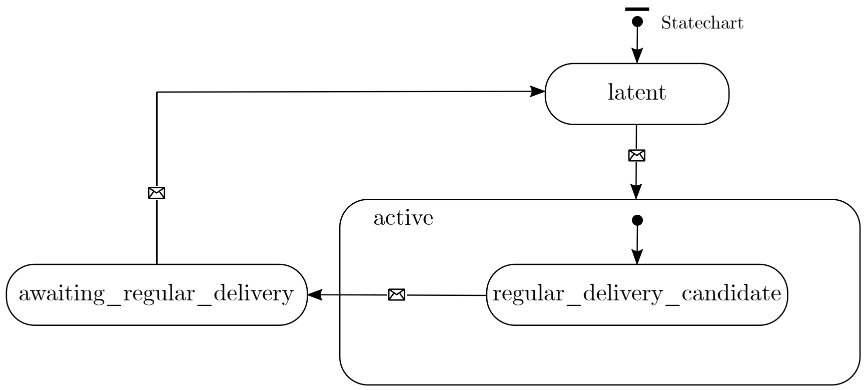


FIGURE 4.1: *The customer agent statechart for model development Phase I.*

All customer agents enter the `latent` state upon initialisation. As soon as an agent places an order, they transition into the `active` state. This state is representative of a customer that has placed an order, but has not yet been assigned to a delivery route. It should be noted that, for Phase I, the `active` state has only a single sub-state, named `regular_delivery_candidate`. The `active` state is, however, extended in Phase III of the model development to cater for a more complex scenario. Once a customer agent is assigned to a delivery route, a message is received by the agent, resulting in a transition to the `awaiting_regular_delivery` state. Finally, once a customer agent's order has been delivered, an additional message is received, and the agent transitions back to the `latent` state. This process may be repeated multiple times for a given customer agent during a simulation run.

The rate at which an individual customer agent enters the `active` state corresponds to the rate at which individual online customers are known to place orders. This rate per customer is modelled as an exponential distribution with a mean value of $\lambda_{customer}$. As the mean order frequency is highly dependent on the specific implementation of the model, relating to the retailer in question, the model user has a degree of control over the assumed mean order rate. This is incorporated by allowing the model user to set the base order rate per customer, which is denoted by λ_0. In a given real-world system, the order rate may be dependent on the time of day. To incorporate the variable demand throughout the day, λ_0 is multiplied by a demand factor that changes throughout the course of a day. The base order rate is captured in the user-defined global parameter `base_order_rate`, whereas the demand factor values are captured in the user defined schedule `daily_demand`. Thus, for a given moment in time, the mean order rate per customer $\lambda_{customer}$ is calculated as

$$\lambda_{customer} = \lambda_0 \times \gamma,$$

where λ_0 and γ respectively denote the base order rate and daily demand factors, captured in the `base_order_rate` parameter and `daily_demand` schedule.

To implement the dynamic placement of orders, the global `place_order` event is introduced in the `main` agent environment. At the occurrence of this event, a message is delivered to a random customer agent in the `latent` state, resulting in their transition to the `active` state. This event occurs at a rate of λ_{system}, which denotes the overall order arrival (as opposed to the rate of an individual customer). For a system with n customer agents, the order arrival rate is calculated as

$$\lambda_{system} = n\lambda_{customer}.$$

The demand schedules can be altered by the model user, based on the specific instance to which it is applied. For demonstration purposes, however, a distinction is made between the demand during normal working hours, defined as 07:00–17:00, after-work hours defined as 17:00–22:00, and hours in which customers are unlikely to place orders, defined to be from 22:00 until 07:00 the next day. The time segments and their associated demand rate values, as captured in the `daily_demand` schedule, are given in Table 4.3.

TABLE 4.3: *The demand factors for daily time segments.*

Time segment	Demand rate factor
07:00–17:00	0.50
17:00–22:00	1.20
22:00–07:00	0.00

4.3.2 Delivery schedule

Deliveries may be performed multiple times per day to serve customers that placed online orders. For each set of deliveries, a number of homogeneous delivery vehicle agents serve all customer agents who ordered before the predetermined cut-off time and are awaiting the delivery of their orders. The delivery process is governed within the `main` agent environment by a schedule, as well as a number of parameters, events, and functions. More specifically, the user-defined parameter `deliveries_per_day` defines the number of regular sets of daily deliveries performed. All sets of deliveries need to start during the retailer's operating hours, as defined by the `depot_open` schedule. The delivery schedules for different values of the `deliveries_per_day` parameter, as modelled, are illustrated in Figure 4.2.

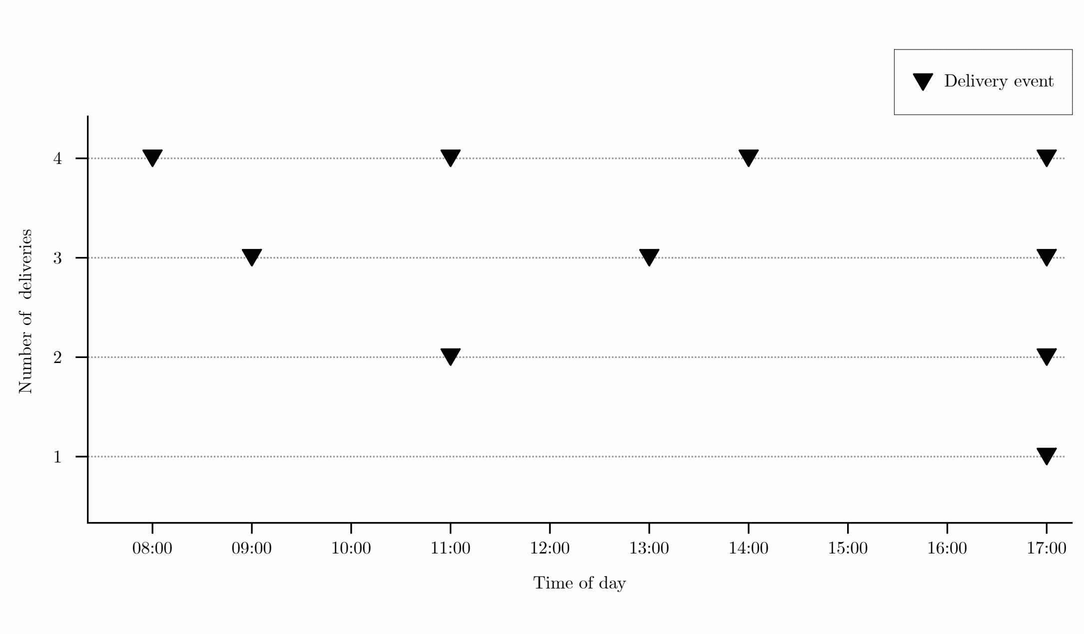


FIGURE 4.2: *The delivery schedule for different number of deliveries per day.*

At these distinct moments in simulated time, the set of deliveries is initiated by the event `perform_deliveries`, which corresponds with the delivery vehicle agents. Prior to the initiation of a set of deliveries, however, retailers may require some time for the preparation and packing of orders for delivery. To account for this time, the event `fix_delivery_routes` occurs 30 minutes before the commencement of the set of deliveries. At the occurrence of this event, all customer agents in the `active` state are selected for the upcoming set of deliveries. A message is delivered to those agents, resulting in a transition from their current state to the `awaiting_regular_delivery` state. All orders placed after the occurrence of the `fix_delivery_routes` event, but before the occurrence of the `perform_deliveries` event, are not considered for delivery as part of the current set, and are only included in the next scheduled set of deliveries.

4.3.3 Delivery vehicle routing

The event `fix_delivery_routes` does not only select the orders for delivery to the respective customers, but also initiates the calculation of the delivery routes. The function `run_cvrp` formulates and solves a CVRP for a given set of customers, ensuring that the demand of each customer is satisfied while adhering to a number of constraints. Thus, to determine the delivery route, the `run_cvrp` function takes as input the locations and demands of awaiting customers, the capacity of the homogeneous delivery vehicles, as well as the location of the depot. These input values are captured in the customer agents in the `awaiting_regular_delivery` state, the parameter `vehicle_capacity`, and the depot agent, respectively. In this section, the complete formulation and solution methodologies of the CVRP are described.

4.3.3.1 CVRP formulation

The general considerations of CVRPs, as discussed in §3.1.1, are taken into account in formulating and solving the CVRP in this instance. Let $\mathcal{C} = \{1, \ldots, n\}$ denote the locations of the set of customers to be served from the central depot (*i.e.* customer agents that are currently in the `awaiting_regular_delivery` state). Furthermore, let the augmented set $\mathcal{N} = \mathcal{C} \cup \{0, n+1\}$ denote the entire set of locations in the network, where 0 and $n+1$ both represent the depot. In particular, 0 represents the depot as a departure point for delivery vehicles, whereas $n+1$ represents the depot as a terminating point for delivery vehicles. Let the demand (as stored in the `demand` parameter of a customer agent) of customer $i \in \mathcal{C}$ be denoted by q_i. Furthermore, let $\mathcal{K} = \{1, \ldots, K\}$ denote the set of homogeneous delivery vehicles, with their capacity (as stored in the parameter `vehicle_capacity` in the `main` agent environment) denoted by Q. As mentioned in §4.2.2.1, it is assumed that K is sufficiently large such that all customers may be served. For a delivery vehicle travelling from location i to location j, a distance of d_{ij} is travelled.

In order to keep track of the movement of delivery vehicles, decision variables are utilised. Let the binary decision variables

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \in \mathcal{K} \text{ travels directly from location } i \in \mathcal{N} \setminus \{n+1\} \\ & \text{to location } j \in \mathcal{N} \setminus \{0\} \\ 0 & \text{otherwise} \end{cases}$$

in order to determine which delivery vehicle is required to visit which subset of customers. Furthermore, let the continuous auxiliary variable u_{jk} denote a lower bound on the cumulative load of vehicle $k \in \mathcal{K}$ before reaching location $j \in \mathcal{N}$. This variable is utilised in breaking subtours, while also considering the capacity of the delivery vehicles.

In line with the conventions typically followed for CVRPs in the literature [65, 124, 189, 191], the model aims to minimise the total cost of delivery. As the cost of delivery is assumed to be directly proportional to the distance travelled by delivery vehicles, the aim is equivalent to minimising the distance travelled. The objective of the CVRP may therefore be formulated as

$$\text{minimise} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{K}} d_{ij} x_{ijk}. \tag{4.1}$$

Furthermore, the model contains a number of standard constraints that are typical of a CVRP. First, to ensure that each customer is visited by a single delivery vehicle, the constraint set

$$\sum_{i \in \mathcal{N} \setminus \{n+1\}} \sum_{k \in \mathcal{K}} x_{ijk} = 1, \quad j \in \mathcal{C} \tag{4.2}$$

is enforced. Furthermore, to ensure that each delivery vehicle departs from the depot, the constraint set

$$\sum_{j \in \mathcal{N} \setminus \{0\}} x_{0jk} = 1, \quad k \in \mathcal{K} \tag{4.3}$$

is enforced. Additionally, the flow conservation constraint set states that any delivery vehicle k arriving at customer j must depart from customer j, for all $j \in \mathcal{C}$. The constraint set

$$\sum_{i \in \mathcal{N} \setminus \{n+1\}} x_{ijk} - \sum_{i \in \mathcal{N} \setminus \{0\}} x_{jik} = 0, \quad k \in \mathcal{K}, j \in \mathcal{C} \tag{4.4}$$

enforces this requirement. Furthermore, all delivery vehicles must terminate their routes at the depot. The constraint set

$$\sum_{j\in\mathcal{N}\setminus\{n+1\}} x_{j,n+1,k} = 1, \quad k \in \mathcal{K} \tag{4.5}$$

ensures adherence to this requirement. Finally, a set of constraints are introduced to eliminate subtours, while also ensuring that the total demand served by delivery vehicle $k \in \mathcal{K}$ does not exceed the maximum capacity of k. The constraint sets

$$u_{ik} - u_{jk} + Qx_{ijk} \leq Q - q_j, \quad i, j \in \mathcal{N}, i \neq j, k \in \mathcal{K}, \tag{4.6}$$

and

$$q_i \leq u_{ik} \leq Q, \quad i \in \mathcal{N}, k \in \mathcal{K} \tag{4.7}$$

enforces adherence to capacity limitations and avoids subtours. Finally, the constraint set

$$x_{ijk} \in \{0, 1\}, \quad i \in \mathcal{N} \setminus \{n+1\}, j \in \mathcal{N} \setminus \{0\}, k \in \mathcal{K} \tag{4.8}$$

is introduced to enforce the binary nature of the decision variables.

4.3.3.2 CVRP solution methodologies

To solve the CVRP formulated in §4.3.3.1, the vast literature on CVRP solution methodologies are consulted. A number of these are discussed in depth in §3.1.1. Due to the substantial number of times that the `run_cvrp` function is executed during a simulation run, the aim is to utilise a solution methodology that produces relatively good solutions within a small amount of time. As such, a heuristic approach is proposed, due to their low computational burden. Two approaches are therefore proposed, namely the savings algorithm by Clarke and Wright [46], as well as the two-phased sweep algorithm by Gillet and Miller [96]. These heuristics are utilised due their low computational burden, ease of implementation, and simple verification.

Savings algorithm

The first methodology described, namely the savings algorithm by Clarke and Wright [46], functions as a constructive heuristic, as described in §3.1.3.1. The first step is determining the list of savings realised by linking two customers. Thus, for the set customers to be served $\mathcal{C}$, let the distance saved by merging customer i and customer j onto a route be calculated as $s_{ij} = d_{0i} + d_{0j} - d_{ij}$ for $i, j \in \mathcal{C}$. Then, let the set $\mathcal{S}$ denote the list of calculated savings, ordered in a non-increasing fashion. Once the list of savings has been determined, the CVRP may be solved constructively by stepping through the list, and making the best feasible merges available. To this end, the algorithm steps through $\mathcal{S}$, making feasible merges until all pairs of linked customers in the list are considered. For a merge to be feasible, the resultant route must adhere to all the constraints described in §4.3.3.1. The basic workings of the heuristic is presented in pseudocode in Algorithm 4.1.

In evaluating each pair of linked customers in $\mathcal{S}$, a number of cases are considered. The first is the case where neither customer i, nor customer j (denoted in Algorithm 4.1 as C_i and C_j, respectively), have been added to a route. If the merge may be feasibly made, these customers are merely linked together to form a new route k. To form this new route, the decision variables x_{0ik}, x_{ijk}, and $x_{j,n+1,k}$ are all set equal to 1.

The second case considers the scenario where customer i is assigned to an extended route, but customer j is not. Furthermore, the extended route must be able to accommodate customer j

Algorithm 4.1: Solving the CVRP using the Clarke and Wright savings heuristic.

Input : A set $\mathcal{S}$ containing the calculated savings s_{ij} for $i, j \in \mathcal{C}$, ordered in a non-increasing fashion.

Output : A set of feasible routes aimed at approximately minimising the total distance required to serve all customers in the set $\mathcal{C}$.

```
for each s_ij in S do
    if C_i and C_j are not assigned to routes AND may be feasibly merged then
        Create new route k
    else if C_i is assigned to a route and C_j is not assigned to a route then
        for each route k in the routes present do
            if C_i is at the start of route k AND C_j can feasibly be added then
                Add C_j to start of route k;
            else if C_i is at the end of route k AND C_j can feasibly be added then
                Add C_j to end of route k;
            else
                continue;
    else if C_i is not assigned to a route and C_j is assigned to a route then
        for each route k in the routes present do
            if C_j is at the start of route k AND C_i may feasibly be added then
                Add C_i to the start of route k;
            else if C_j is at the end of the route AND C_i may feasibly be added then
                Add C_i to the end of route k;
            else
                continue;
    else if C_i and C_j are both assigned to routes then
        for each route k in the routes present do
            for each route l in the routes present do
                if C_i and C_j are not on the same route AND can be feasibly merged then
                    if C_i is first on route k and C_j is first on route l then
                        Reverse(route k);
                        Add route l to route k;
                    else if C_i is last on route k and C_j is last on route l then
                        Reverse(route l);
                        Add route l to route k;
                    else if C_i is first on route k and C_j is last route l then
                        Add route k to route l;
                    else if C_i is last on route k and C_j is first on route l then
                        Add route l to route k;
                    else
                        continue;
                else
                    continue;
end;
```

without breaking any constraints. In such a case, two feasible scenarios exist where the customers may be merged onto the same route. If customer i is the first customer on a particular route k (*i.e.* $x_{0ik} = 1, x_{i,n+1,k} = 0$), then customer j may be added to the route before customer i. This action is performed by setting the decision variable x_{0ik} equal to 0, while setting x_{0jk} and x_{jik} equal to 1. Alternatively, if customer i is the last customer on its route (*i.e.* $x_{0ik} = 0, x_{i,n+1,k} = 1$), then customer j may be added to the route after customer i. This action is performed by setting the decision variable $x_{i,n+1,k}$ equal to 0, while setting $x_{j,n+1,k}$ and x_{ijk} equal to 1. If customer i is neither at the start nor at the end of the route, then no merge is made and the pair of customers associated to the next savings value is considered.

The third case considers the opposite scenario to the second case, where customer j is assigned to an extended route, but customer i is not. Furthermore, the extended route must be able to accommodate customer i without breaking any constraints. Similarly, two feasible scenarios exist where the customers may be merged onto the same route. If customer j is the first customer on its route (*i.e.* $x_{0jk} = 1, x_{j,n+1,k} = 0$), then customer i may be added to the route before customer j. This action is performed by setting the decision variable x_{0jk} equal to 0, while setting x_{0ik} and x_{ijk} equal to 1. Alternatively, if customer j is the last customer on its route (*i.e.* $x_{0jk} = 0, x_{j,n+1,k} = 1$), then customer i may be added to the route after customer j. This action is performed by setting the decision variable $x_{j,n+1,k}$ equal to 0, while setting $x_{i,n+1,k}$ and x_{jik} equal to 1. If customer j is neither at the start nor at the end of the route, then no merge is made and the pair of customers associated to the next savings value is considered.

Finally, the case is considered where both customer i and customer j have been assigned to extended routes. Moreover, it only considers cases where customer i and customer j are on separate routes, which may be feasibly merged into a single route. This includes four scenarios where neither customer i nor customer j are in the middle of their respective routes. In the first scenario, customer i is first on route k, whereas customer j is first on route l. In this scenario both of the paths from the depot are removed by setting the decision variables x_{0ik} and x_{0jl} equal to 0, while a path connecting the two customers is added. One of the routes, however, needs to be reversed in order for a feasible merge to be possible. As such, route k is reversed before appending route l to the end.

Similarly, in the scenario where customer i is last on route k and customer j is last on route l (*i.e.* $x_{0ik} = x_{0jl} = 0, x_{i,n+1,k} = x_{j,n+1,l} = 1$), the two may be feasibly merged after reversing one of the routes and removing the respective paths to the depot. As such, route l is reversed before route k is appended by adding a path from customer i to customer j and removing their respective paths linking them to the depot. In the third scenario, customer i is first on route k, while customer j is last on route l (*i.e.* $x_{0ik} = x_{j,n+1,l} = 1, x_{i,n+1,k} = x_{0jl} = 0$). The customers may be merged by adding route k to the end of route l, and removing the respective paths linking customers i and j to the depot. Finally, in the scenario where customer i is last on route k, while customer j is first on route l (*i.e.* $x_{0ik} = x_{j,n+1,l} = 0, x_{i,n+1,k} = x_{0jl} = 1$), route k is extended to include route l by joining customer i and customer j, while removing their respective links to the depot.

Sweep algorithm

The second methodology considered, namely the algorithm by Gillet and Miller [96], functions as a cluster first route second approach, as described in §3.1.3.2. In the first phase, a set of capacitated clusters are formed, with each cluster associated to a particular delivery vehicle. Thereafter, each cluster is treated as a TSP and the route of each delivery vehicle is solved individually.

In the first phase, clusters are formed based on the distribution of the customers around the depot. To this end, let (x_i, y_i) denote the coordinates of location $i \in \mathcal{N}$ that is visited by the delivery vehicles. Furthermore, let θ_i denote the angle of customer agent $i \in \mathcal{C}$ relative to a chosen reference line, calculated as

$$\theta_i = (90^\circ - \arctan \frac{y_i - y_h}{x_i - x_h}) \mod 360^\circ.$$

The selected reference line has an origin at the depot and points in the positive y-direction. The calculation of θ_i is illustrated for two customers in Figure 4.3.

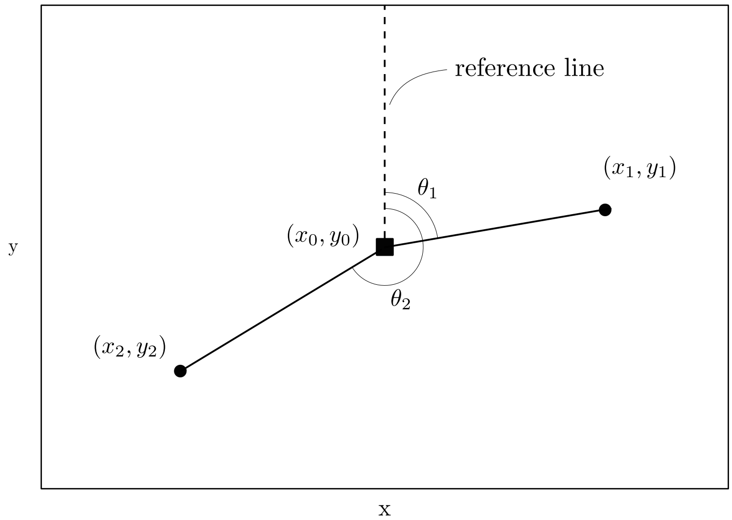


FIGURE 4.3: *An illustration of the calculation of angles of two customers for the sweep heuristic.*

Let the customers be sorted in an increasing order of their θ value, such that $\theta_i < \theta_{i+1}$ for all $i \in \mathcal{C}$. This ordered set is used as input to the capacitated clustering phase of the heuristic. This phase is described in pseudocode in Algorithm 4.2. The algorithm sequentially assigns customers to a delivery vehicle until its capacity is reached, before assigning customers to the next delivery vehicle. This process is repeated until all customers have been assigned to a vehicle. The resultant output is K clusters, where each cluster $\mathcal{R}_k$ contains the customers to be visited by the delivery vehicle k.

Algorithm 4.2: The sweep vehicle routing heuristic — Cluster phase.

Input : A set $\mathcal{C}$ containing n customer agents, sorted in increasing order of their θ value, with demand values q_i; delivery vehicle capacity Q.

Output: K clusters $\mathcal{R}_k$ containing the customers for each vehicle $k \in \mathcal{K}$.

$i \leftarrow 1$; `// start at first customer`
for *DeliveryVehicle* $\leftarrow 1$ ***in*** $\mathcal{K}$ **do**
  vehicleLoad $\leftarrow 0$; `// set current vehicle load to 0`
  while *vehicleLoad* $\leq Q$ **do**
    vehicleLoad $\leftarrow$ *vehicleLoad* $+ q_i$;
    add customer i to $\mathcal{R}_k$;
    $i \leftarrow i + 1$;
    if $i > n$ **then break**

In the second phase of the sweep heuristic, each cluster $\mathcal{R}_k$ for $k \in \mathcal{K}$ is treated as a TSP and solved individually. Although various solution methodologies exist for solving a TSP, a heuristic approach is taken for its relatively good solutions in a short amount of time. In particular, each individual TSP is solved using the savings heuristic, as described in Algorithm 4.1. It is known, however, that the cumulative demand of each cluster $\mathcal{R}_k$ does not exceed the capacity of vehicle k. As such, the capacity constraint need not be considered while executing the algorithm in forming the solution of the TSP. For a certain cluster $\mathcal{R}_k$, let the distance saved by merging customer i and customer j onto a route be calculated as $s_{ij} = d_{0i} + d_{0j} - d_{ij}$ for $i, j \in \mathcal{R}_k$. Then, let the set $\mathcal{S}_k$ denote the list of savings associated with the cluster $\mathcal{R}_k$, ordered in a non-increasing fashion. Then step through $\mathcal{S}_k$, making feasible merges until a single route is formed. As such, for the second phase of the sweep algorithm, Algorithm 4.1 is executed k times, considering the customers in $\mathcal{R}_k$ (rather than in $\mathcal{C}$) while stepping through savings list $\mathcal{S}_k$ (rather than $\mathcal{S}$).

Irrespective of the choice of routing heuristic, the resultant delivery routes are returned by the `run_cvrp` function. Furthermore, the function returns the distances to be travelled by each delivery vehicle agent. In order to determine the associated cost of delivery, the user-defined input parameter `cost_per_distance` defines the cost per unit distance travelled by a delivery vehicle agent.

Each delivery vehicle agent has a collection `delivery_route`. This collection is populated with the customer agents to be visited by the delivery vehicle agent at each execution of the `fix_delivery_routes` event, and subsequently cleared at the completion of the delivery event.

4.3.4 Delivery vehicle agent

The statechart of the `delivery_vehicle` agent population is shown in Figure 4.4. Agents in the `at_depot` state wait passively at the depot until the commencement of a set of deliveries. At the occurrence of the event `perform_deliveries`, a message is delivered to all delivery vehicle agents, triggering a transition to the `active` state. A transition guard ensures that only delivery vehicle agents with an assigned route (*i.e.* a non-empty `delivery_route` collection) are able to transition. This emulates the simultaneous start of deliveries by all delivery vehicles required for the current set of deliveries.

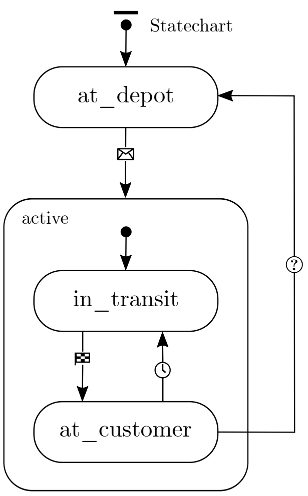


FIGURE 4.4: *The statechart illustrating the behaviour of a delivery vehicle agent.*

Within the `active` state, the delivery vehicle agent visits each of the customer agents in its `delivery_route` collection sequentially, removing a customer agent from the collection once the delivery to that agent is complete. More specifically, once in the `in_transit` state, the delivery vehicle agent travels at a speed of 40 kilometres per hour to the first customer agent in its `delivery_route` collection. As soon as the customer agent is reached, the delivery vehicle agent transitions to the `at_customer` state and, according to a time-out, the agent transitions back to the `in_transit` state, unless no further customer agents remain to be served. The timeout simulates the time spent at the online customer's home location and is set to a duration of five minutes. Once the delivery of an order is completed, a message is delivered to the customer agent being served, triggering their transition back to the `latent` state within the `customer` agent statechart (as depicted in Figure 4.1) and removing them from the `delivery_route` collection of the delivery vehicle agent. Once the final customer is served (*i.e.* the `delivery_route` collection is empty), the conditional transition is triggered. The delivery vehicle agent transitions to the `at_depot` state and travels back to the location of the depot agent.

4.4 Phase II — Occasional driver behaviour

In the second phase of the model development, the behaviour of ODs (*i.e.* the in-store customer that has the option of performing deliveries to online customers) are considered. This phase of the model development is therefore only concerned with the `OD` agent population, detailing the arrival rate of OD agents, as well as their general movements. Specific focus, however, is placed on describing the manner in which ODs associate value with their travel time and the associated modelling approach. This is imperative to understanding the integrated customer crowd-shipping system, as described in Phase III.

The statechart of the `OD` agent population is displayed in Figure 4.5 and governs the behaviour of OD agents. OD agents are initialised in the `at_home` state and placed at their home location, defined by Cartesian coordinates as captured in the agents' `home_coordinates` parameter. During model execution, an OD agent may enter the `shopping` state, governed by a rate transition associated with the OD arrival rate. Upon entering this state, the OD agent will travel from their home location to the location of the depot agent, in order to mimic the shopping for goods, before returning home. When an OD agent leaves their home location, a list of potential online order deliveries, along with their associated incentives, may be presented to them for selection. This process emulates the functionality of a crowdsourcing platform, such as a mobile application, in a real-world customer crowd-shipping initiative. An OD may receive a notification when new offers become available, or view all available offers on the crowdsourcing platform before leaving home. If the OD decides to deliver a particular order, their selection may be communicated to the retailer through the crowdsourcing platform. The selection is modelled to be based on self-interest, with all OD agents aiming to maximise their own perceived utility derived from the selection. If none of the offers are perceived as beneficial, the OD agent rejects all offers and returns to the `at_home` state after visiting the depot. On the other hand, if an offer is accepted, the OD agent transitions to the `delivering_order` state after visiting the depot and travels to the location of the associated customer agent. Once the delivery is completed, the OD agent transitions back to the `at_home` state and travels back to their home location.

The number of OD agents in the system is governed by the parameter `OD_customer_ratio`, which may be defined by the model user. This parameter essentially captures the popularity or maturity of the initiative by defining the number of OD agents as a fraction of the number of online customer agents in the system. As an example, if there are 100 customer agents in the

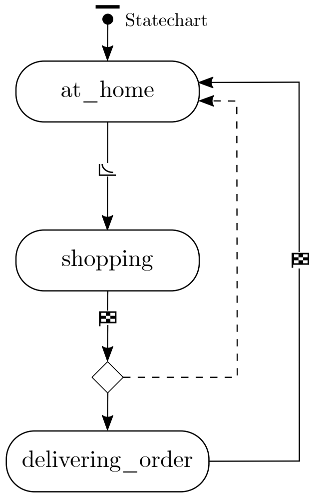


FIGURE 4.5: *The statechart of an occasional driver agent.*

system, an `OD_customer_ratio` parameter value of 1.2 would result in 120 OD agents in the system.

4.4.1 Arrival rate

The frequency at which an OD agent travels to the location of the depot is captured in the rate transition from the `at_home` state to the `shopping` state. The rate of this transition is modelled as an exponential distribution with a mean denoted by the `base_OD_rate` parameter. This user-defined parameter, in combination with the `OD_customer_ratio` parameter, governs the number and frequency of ODs visiting the depot. It is further modelled that OD agents are only able to transition into the `shopping` state during times when the depot is open (*i.e.* when the `depot_open` schedule is set to TRUE).

4.4.2 Value of time

When acting as an OD, an individual spends time, in addition to their original journey time, to perform a delivery. As mentioned in §3.4, individuals often perceive the time spent in urban commuting trips as more costly than the associated vehicle operating costs. As such, an OD's willingness to perform a delivery is dependent on the opportunity cost of the additional required travel time. For an OD to be willing to perform a delivery, the compensation offered for the trip should be greater than the individual's perceived value of the additional travel time. It is therefore necessary to model the perceived monetary value associated with any given amount of additional travel time for an individual.

4.4.2.1 Individual value of time

For an individual OD agent, the monetary value of performing a delivery is dependent on the perceived additional amount of time required to complete the delivery, as well as their willingness to trade travel time for money. The modelling of these elements are discussed subsequently.

The first consideration is the perceived additional time required for an OD agent to deliver an order to a customer agent. To describe the modelling approach, consider the theoretical case where an OD agent is incentivised to deliver an order to a customer agent, with δ denoting the deviation in distance required. The OD agent, however, evaluates the trip not primarily on the distance deviation, but rather on the additional time required. The `average_speed` parameter captures an OD agent's estimated speed for the duration of a trip. The additional time required, as perceived by the OD agent, is then calculated as

$$t = \frac{\delta}{\texttt{average\_speed}}.$$

As mentioned in §3.4, the subjective value of time may be modelled by considering an individual's MRS between travel time and travel cost. Thus, let MRS_{vt} denote an OD agent's willingness to trade an additional unit of travel time for a unit of money. Then, the monetary value associated with delivering a particular order, denoted by v, may be modelled as a function of the perceived additional travel time required, denoted by t, as well as the MRS_{vt} of the particular OD. This may be denoted as

$$v = f(t, MRS_{vt}).$$

For the purpose of this model, a simplifying assumption is made which states that an OD agent's willingness to trade travel time for money may be modelled as perfect substitutes. This implies that their MRS of time and money is constant, independent of the amount of time or money. By making this assumption, the MRS_{vt} of each OD agent may be captured as a constant value in their `base_value_of_time` parameter. Thus, a particular OD agent's associated monetary value for delivering an order is calculated as

$$v = t \times \texttt{base\_value\_of\_time}.$$

This concept is illustrated in Figure 4.6, where the value of time curve for an individual OD agent is plotted over the additional travel time required. The curve has a slope equal to the `base_value_of_time` parameter value for the OD agent in question. As mentioned in §3.4.1, an individual's willingness to accept a longer trip duration in exchange for a lower cost is referred to as their WTA. The customer crowd-shipping scenario creates a unique case of WTA, where a longer trip (*i.e.* the additional travel time t) is essentially exchanged for a lower travel cost (as a result of the offered incentive). The value of v may therefore be interpreted as an individual's WTA (*i.e.* the point at which they are indifferent between their original trip and receiving v monetary value as compensation for travelling t additional time). Therefore, as illustrated in Figure 4.6, for a trip requiring an additional travel time of t_{ref}, the OD agent would require an incentive greater than v_{ref} to perform the delivery in question.

4.4.2.2 Factors influencing value of time

In modelling independent agents, the value of additional travel time may vary throughout the OD agent population. As mentioned in §3.4.3, a number of factors may be indicative of how highly an individual values their time. Two of these factors are incorporated into the model, resulting in variability across the population of OD agents. This includes the individual's income level and the presence of congestion during the time at which the trip is taken.

The income of the OD agent population is modelled as three distinct income brackets, namely low, middle, and high income. Each income bracket is associated with a different range within which the `base_value_of_time` parameter value may fall. In Figure 4.7, the value of time

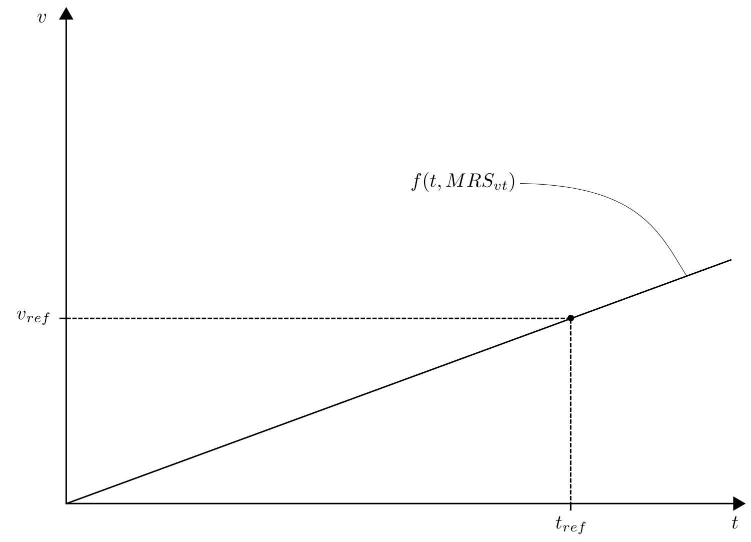


FIGURE 4.6: *The value of time curve for an individual OD agent.*

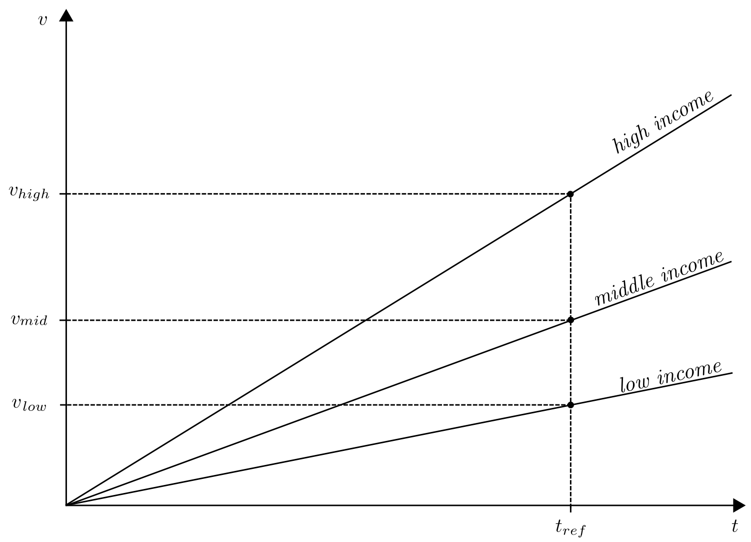


FIGURE 4.7: *The value of time curves for three OD agents, each representative of a different income bracket.*

curves for three individuals, each representative of a different income bracket, are shown to illustrate the slope differences.

A steeper slope and higher value of the `base_value_of_time` parameter is associated with individuals in a higher income bracket. For a given additional travel time of t_{ref}, individuals with a low, middle, and high income will require a compensation greater than v_{low}, v_{mid}, and v_{high}, respectively, to be willing to deliver the order. Each OD agent has a parameter `income_level`, which may take a value of 0, 1, or 2, respectively representing a low, middle, or high income bracket. To emulate a particular income distribution for the OD agent population, the value of the `income_level` parameter is probabilistically assigned to each OD agent upon its initialisation. As such, let p_{low}, p_{mid}, and p_{high} denote the probability of an agent being assigned an `income_level` parameter value of 0, 1, or 2, respectively. These probabilities are captured in the custom distribution `income_bracket_probabilities` in the `main` agent environment, which ultimately influences the OD population's value of time distribution.

Instead of having a deterministic MRS value for all OD agents within a certain income bracket, the `base_value_of_time` parameter values are distributed normally per income bracket. More formally, the value of time of an individual is distributed such that

$$\texttt{base\_value\_of\_time} = \text{Normal}(\mu_b, \sigma_b)$$

where μ_b and σ_b respectively denote the mean and variance of the modelled income bracket b.

In Figure 4.8, the distribution of the MRS values according to different income brackets are illustrated. The mean MRS values for the low, middle, and high income brackets are denoted by μ_{low}, μ_{mid} and μ_{high}, respectively. Similarly, the variance of these distributions are denoted by σ_{low}, σ_{mid} and σ_{high}. Thus, for a given OD agent with a `income_level` parameter value equal to 2, the agent's `base_value_of_time` parameter value is drawn from the distribution Normal($\mu_{high}, \sigma_{high}$). Moreover, it can be noted that for this example the probability density decreases from the low income bracket to the middle and high income brackets. This provides insight into the `income_bracket_probabilities` distribution, indicating that $p_{low} > p_{mid} > p_{high}$.

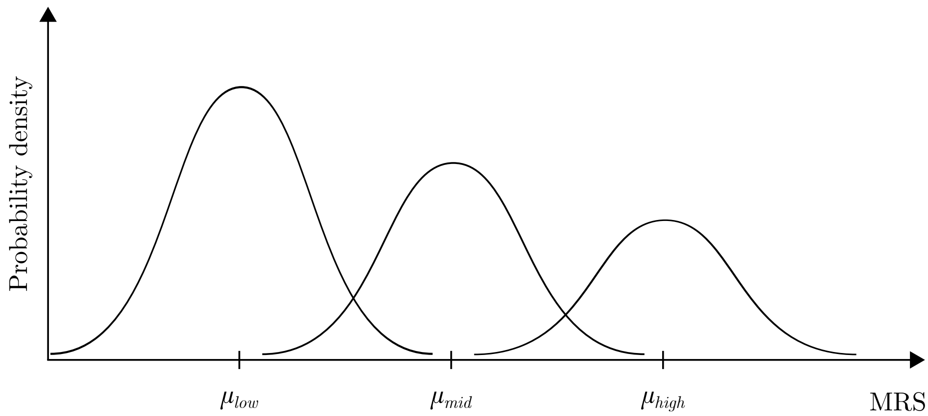


FIGURE 4.8: *The probability distributions of MRS values for the different income brackets.*

Finally, the influence of congestion on the valuation of time is considered. Peak congested hours are defined by the schedule `peak_hours`, which takes the value of 1 in regular hours and a user-defined value greater than 1 during the defined peak hours. In adjusting the MRS value to incorporate the effect of congestion, the `base_value_of_time` parameter value is multiplied by the value associated with the specific hour, as stored in the `peak_hours` schedule, each time an OD agent enters the `shopping` state. It is assumed that agents from different income brackets are influenced by congestion similarly. This shift in MRS as a result of congestion is illustrated in Figure 4.9.

Thus, to ensure that the value of time for an OD agent adjusts dynamically due to congestion, the agent's value of time is recalculated each time they enter the `shopping` state. The variable `value_of_time` captures the updated value of time of an OD agent and is calculated as

$$\texttt{value\_of\_time} = \texttt{base\_value\_of\_time} \times \texttt{peak\_hours}.$$

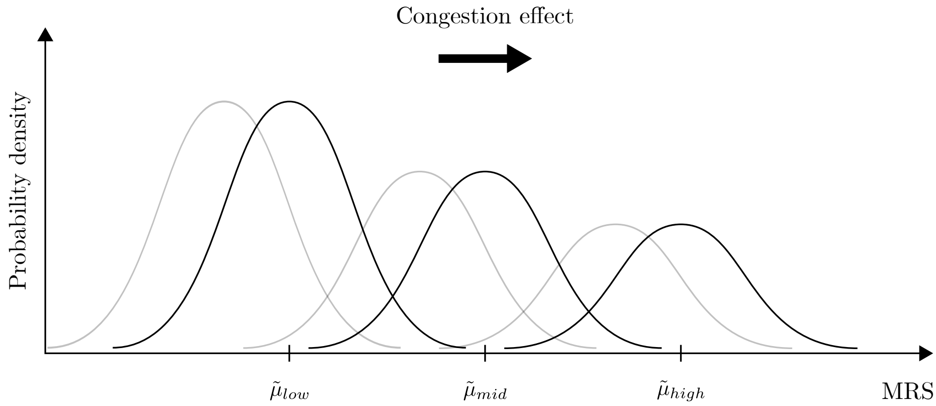


FIGURE 4.9: *The effect of congestion on the MRS distributions for the various income brackets.*

4.4.2.3 Model parameter values

For the implementation of value of time in this model, the model user defines μ_{low}, the mean value of the MRS distribution for the low income bracket. This value is captured in the `minimum_VOT` global parameter. Moreover, in order to provide adequate variation throughout the `OD` agent population, the mean MRS values of the middle and high income distributions are calculated as a factor of the `minimum_VOT` parameter. These constant factors are calibrated to provide sufficient diversity across the income brackets. Specifically, the mean MRS values of the middle and high income distributions are then calculated as $\mu_{mid} = (1.6)$ `minimum_VOT` and $\mu_{high} = (2.2)$ `minimum_VOT`, respectively. Similarly, the standard deviation of an MRS distribution provides further diversity within the income bracket. Upon inspection, a standard deviation value of 0.4 was deemed as providing sufficient diversity within each income bracket. As such, it is assumed that the standard deviation of the given MRS distributions are equal, with $\sigma_{low} = \sigma_{mid} = \sigma_{high} = 0.4$. Finally, the peak congestion intervals and the associated values, as captured in the `peak_hours` schedule, are shown in Table 4.4.

TABLE 4.4: *The time value factors as per the congestion schedule.*

Time segment	Time value factors
08:00–09:00	1.50
09:00–16:00	1.00
16:00–17:00	1.50
17:00–08:00	1.00

4.5 Phase III — Integrated approach to customer crowd-shipping

The final phase of the model development involves integrating the customer crowd-shipping initiative into the retailer's last-mile delivery system. The company seeks to outsource the fulfilment of unwanted deliveries by presenting these as an open call to the OD agents. The ODs, in turn, act as autonomous agents in deciding whether or not to accept an offered task. The overarching goals of the customer crowd-shipping initiative is to encourage participation from ODs, while reducing both the cost of last-mile deliveries, as well as the customer waiting time.

As such, the key factors to consider include determining which customer agents to outsource such that it will reduce the company's delivery vehicle costs. Furthermore, an incentive scheme should be developed such that the reduction in delivery vehicle costs are not overshadowed by the price of the incentives offered to outsource the deliveries. The incentive should, however, be of a value such that a sufficient number of OD agents choose to perform the delivery tasks presented.

To achieve this, a methodology is proposed to dynamically calculate the cost-to-serve values of all customer agents in the system, while an incentive scheme is proposed to encourage ODs to perform deliveries. Furthermore, a methodology is devised for proposing orders for crowd-shipping, based on the cost-to-serve of the particular customer agent, a user-defined loss aversion parameter, and the values of the incentive scheme. Finally, the modelled decision making of an OD agent is described, based on the value of the incentive offered and the value of the additional time required to deliver an order (*i.e.* their WTA an extended trip).

4.5.1 Cost to serve

In order to determine which online customers should be identified candidates for crowd-shipping, it is important to consider the cost of serving each specific customer by means of the traditional scheduled set of deliveries. Since delivery routes include multiple customers that are served by a number of capacitated delivery vehicles, the cost-to-serve value of a customer cannot merely be calculated as the distance between the depot and the customer. Rather, the cost-to-serve value is defined as the difference between the cost of the routes with and without a particular customer agent. The choice of routing algorithm is therefore essential to this calculation and it is assumed that the delivery vehicles will adhere to the routes provided by the selected VRP solution methodology. The pseudocode presented in Algorithm 4.3 describes the calculation of the cost-to-serve value c_i of each customer $i \in \mathcal{C}$, while utilising a given routing algorithm (such as those described in §4.3.3.2), where R_{max} denotes the routing cost for the entire set of customers and R_i represents the routing cost without customer i.

Algorithm 4.3: Calculate cost-to-serve

Input : A set $\mathcal{C}$ containing n customers to visit.
Output: A cost-to-serve value c_i for each customer i.

```
R_max ← RoutingAlgorithm(C);
for i ← 1 to n do
    R_i ← RoutingAlgorithm(C \ {i});
    c_i ← R_max − R_i;
```

The function `calculate_cost_to_serve` calculates the cost-to-serve values of all customer agents in the `active` state utilising one of the routing heuristics described in §4.3.3.2. The function therefore initially returns the cost-to-serve as distance-based values, which is subsequently converted to a monetary value by multiplying the values by the `cost_per_distance` global parameter. The resultant values are then captured in the `cost_to_serve` variable for each customer agent.

4.5.2 Incentive scheme

The incentives scheme employed in the model comprises three components, namely a fixed incentive, a variable rate, and a maximum incentive, denoted by λ_{fixed}, λ_{rate}, and λ_{max}, respectively. For each potential instance of customer crowd-shipping, a fixed amount of λ_{fixed} is offered, regardless of the deviation required by the OD agent. Additionally, a distance-based rate λ_{rate} is offered for each additional unit of distance travelled by the OD agent as they deliver the order before returning home. Finally, to avoid instances where exorbitant offers are proposed by the retailer, λ_{max} defines the upper limit for the monetary value that may be offered.

More formally, for OD agent j delivering to customer agent i by making a deviation of δ_{ij}, the total incentive is calculated as

$$\lambda_{ij} = \min(\lambda_{fixed} + \delta_{ij}\lambda_{rate}, \lambda_{max}). \tag{4.9}$$

This methodology for calculating incentives implies that the retailer has knowledge of the OD's destination. As mentioned in §3.5.4, the verification of an OD's destination may be a significant challenge in customer crowd-shipping and of particular importance when incorporated into the calculation of the offered incentive. ODs may attempt to exploit the incentive scheme by declaring a false destination, resulting in an exorbitantly high incentive. The introduction of a maximum incentive serves as a mitigating factor, limiting the monetary value of an incentive offered. In addition to this, the use of a crowdsourcing platform may further deter such fraudulent behaviour. As mentioned in §2.3.1, online platforms enable collaborative business models through the collection and utilisation of data over time. For example, the requirement that ODs declare their home location upon registering for the initiative may serve as a critical risk reducing factor. Moreover, the knowledge of an OD's past transactions and behaviours may establish a track record, which may cultivate trust and, ultimately, improve collaboration.

The incentive parameters λ_{fixed}, λ_{rate}, and λ_{max} are captured in the global parameters, namely `fixed_incentive`, `variable_rate`, and `max_incentive`, respectively. These are user-defined parameters, defined in the `main` agent environment.

4.5.3 Selecting candidate orders for crowd-shipping

The aim of the customer crowd-shipping initiative is to utilise ODs in cases where regular last-mile deliveries are expensive. As such, in the proposed model, a customer agent is classified as a candidate for crowd-shipping based on its `cost_to_serve` variable value. The aim is therefore to outsource customer agents with a `cost_to_serve` variable value greater than the value of the incentive payable to the OD agent. This cost of outsourcing, however, is dependent on knowledge of the particular OD agent to be utilised and their required deviation in advance. The actual cost of outsourcing a particular customer therefore remains unknown until a particular OD agent accepts the offer. As such, it is not possible to determine the cost of outsourcing a particular customer pre-emptively. It is certain, however, that the incentive paid will be at least equal to the `fixed_incentive` parameter value. Thus, in the case where a customer agent has a `cost_to_serve` variable value smaller than the `fixed_incentive` parameter value, outsourcing the customer would invariably result in a loss. A threshold for classifying a customer agent as a crowd-shipping candidate is therefore set as a factor of the fixed incentive offered to ODs. Specifically, any customer agent for which the inequality

$$\texttt{cost\_to\_serve} > \beta\,\texttt{fixed\_incentive} \tag{4.10}$$

holds, is classified as a crowd-shipping candidate, with $\beta \geq 0$ denoting a loss aversion parameter. A β value of 1 implies that customer agents with a `cost_to_serve` variable value equal or less than the `fixed_incentive` parameter value are to be served by the regular deliveries. Thus, the cases where losses are guaranteed are eliminated. Furthermore, a small value of β indicates a liberal approach to outsourcing customers, where crowd-shipping is encouraged even if a regular company delivery may have been inexpensive. On the other hand, a large value of β corresponds to a conservative approach, where customers are only candidates for crowd-shipping if the cost of regular delivery is extremely high. This approach does not guarantee savings in each instance of crowd-shipping, but rather aims to eliminate the instances where losses are guaranteed. The β parameter is captured in the `loss_aversion` parameter and may be varied by the model user to reflect the retailer's desired approach to selecting candidates for crowd-shipping.

The initial statechart of the `customer` agent population, as presented in Figure 4.1, is expanded to accommodate the incorporation of customer crowd-shipping. The full statechart of the `customer` agent population is shown in Figure 4.10. Once a customer agent places an order (thereby transitioning from the `latent` to the `active` state) and enters the `unassigned` state, the `calculate_cost_to_serve` function is executed for all customer agents in the overarching `active` state. The `cost_to_serve` variable values of all customer agents that have ordered, but have not yet been assigned to a route, are therefore updated to include the newly active customer agent on the planned route.

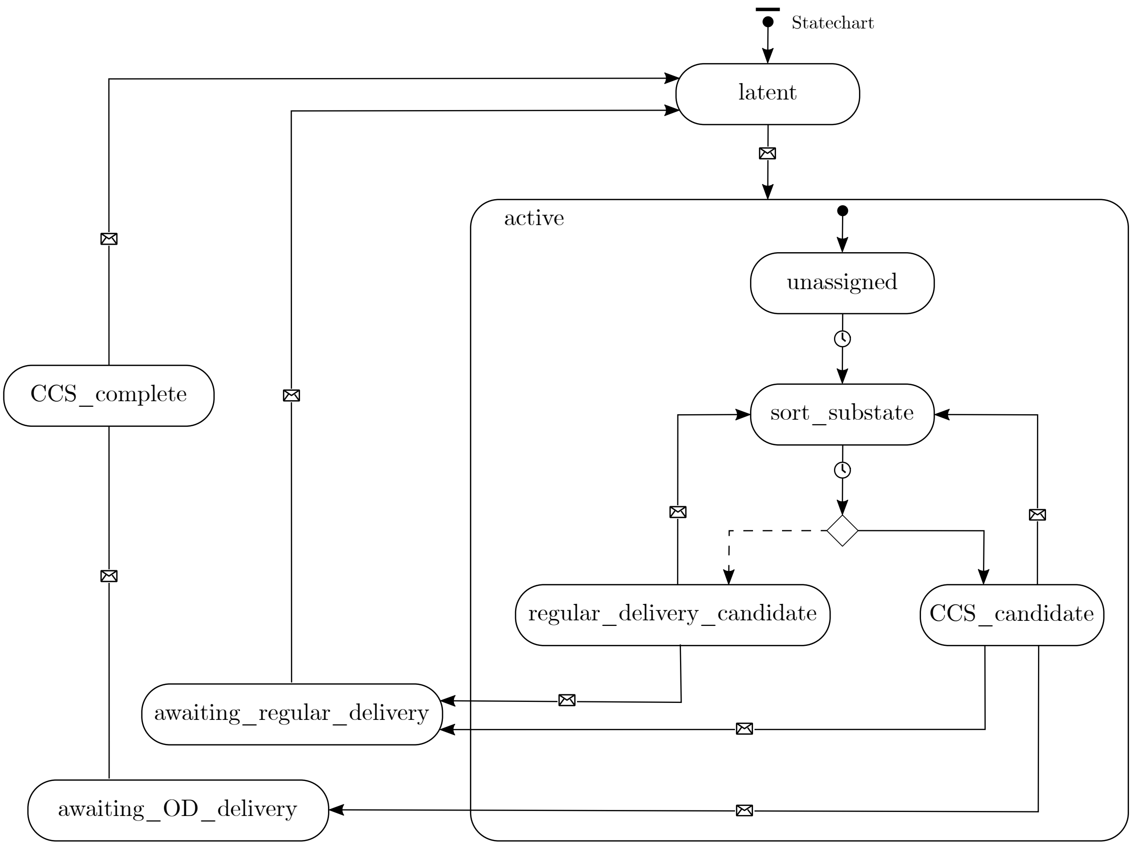


FIGURE 4.10: *The statechart depicting the behaviour of a customer agent.*

After the `cost_to_serve` variable values are updated, the customer agent transitions to the `sort_substate`, before being classified as a candidate for crowd-shipping or for regular delivery. If the inequality in 4.10 holds, the customer agent enters the `CCS_candidate` state, otherwise they enter the regular_delivery_candidate state. Customer agents in the CCS_candidate state are presented as crowd-shipping options to OD agents each time they intend to visit the store, whereas customer agents in the regular_delivery_candidate state are only served once the company performs last-mile deliveries.

As new orders arrive, the cost_to_serve variable values of active customers are updated. As such, it is necessary to re-sort the active customer agents into their appropriate state each time a new order arrives. Thus, each time the calculate_cost_to_serve function is executed, a message is sent to all customer agents in the CCS_candidate and regular_delivery_candidate states. This message results in a transition to the sort_substate, whereafter they are re-sorted based on their updated cost_to_serve variable values and expression in 4.10.

Customer agents in the CCS_candidate state may be served by willing OD agents, resulting in their transition to the awaiting_OD_delivery state, before further transitioning to the CCS_complete state (this process is discussed subsequently). Alternatively, any customer agents in the CCS_candidate state that have not yet been served by an OD agent by the time a set of regular deliveries is fixed, are added to a regular delivery route. As described in §4.3.2, the routes for the regular set of deliveries are fixed at the execution of the fix_delivery_routes event. This event broadcasts a message to all customer agents in the active state, selecting them for the upcoming set of deliveries. Specifically, customer agents in both the regular_delivery_candidate and CCS_candidate states are selected and transition to the awaiting_regular_delivery state. Once the delivery is complete, the customer agent transitions back to the latent state.

In more simple terms, customer agents in the CCS_candidate state may be served by OD agents throughout their waiting period. If they have not yet been served by the time the delivery routes are fixed, they are added to the set of regular deliveries. Customer agents in the regular_delivery_candidate state, on the other hand, merely wait for the set of regular deliveries. Thus, in every set of regular deliveries, all customer agents are served, regardless of their state before the execution of the fix_delivery_routes event.

4.5.4 OD decision-making process

Once an OD agent transitions from the at_home to the shopping state (as depicted in Figure 4.5), the function evaluate_CCS_candidates is executed. This function models the decision making of ODs as they are presented with a list of potential online orders to fulfil. The list of potential orders presented to the OD agent include that of customer agents in the CCS_candidate state. The OD agent is presented with the locations of the online customer agents, and the incentive offered for delivering each of these orders. Ultimately, the OD agent will evaluate their options based on the perceived value of the additional time compared to the value of the incentive, and select the offer with the highest perceived gain. If no offer is perceived as beneficial, the OD may reject all offers.

Suppose an OD agent j has the option of delivering to customer agent i by making a deviation of δ_{ij}, with an estimated additional time of t_{ij} required and is offered a compensation of λ_{ij}. The value of the additional time is denoted by v_{ij}, and the perceived gain or loss is then calculated as

$$g_{ij} = \lambda_{ij} - v_{ij}. \tag{4.11}$$

This is shown graphically in Figure 4.11, illustrating the value of time function of OD agent j, as well as the offered incentive λ_{ij}. After evaluating all available offers, the OD agent selects the order with maximum perceived gain.

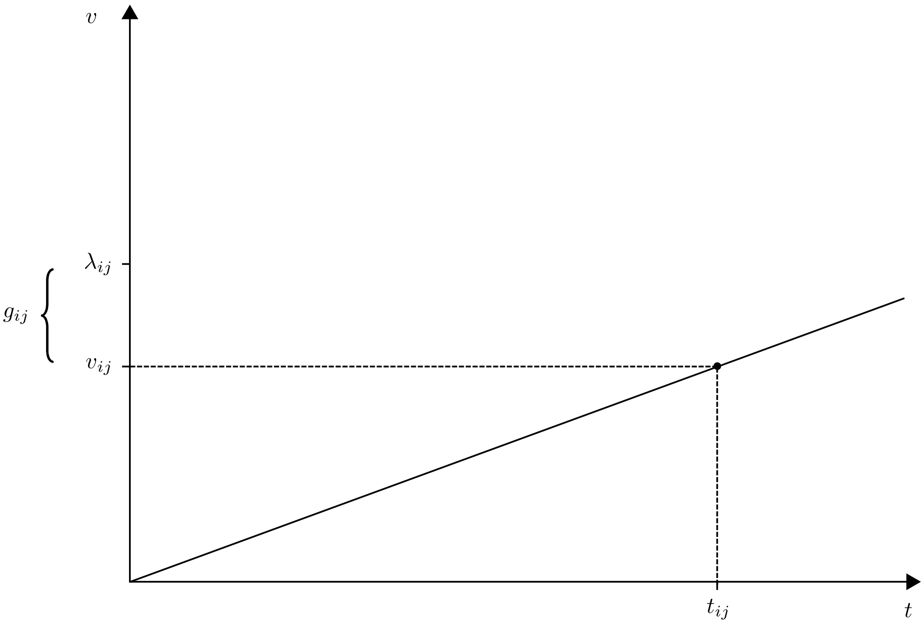


FIGURE 4.11: *An illustration of the perceived gain for an individual customer agent according to the incentive offered.*

If an OD agent has accepted the offer to deliver an order to a certain customer agent, the OD agent's variables `selected_order` and `accepted_incentive` temporarily captures the identity of the selected customer agent and the associated incentive offered, respectively. The OD agent transitions from the `shopping` state after arrival at the depot. If the `selected_order` is empty (*i.e.* no offer has been accepted), the OD transistions to the `at_home` state and travels back to their home location. On the other hand, if the `selected_order` variable contains a selected customer, the OD agent transitions to the `delivering_order` state and transmits a message to the `selected_order` customer agent. The message triggers the selected customer agent to transition from the `CCS_candidate` state to the `awaiting_OD_delivery` state (depicted in Figure 4.10). This process emulates the selection of the offer on a crowdsourcing platform. As such, the associated customer agent is subsequently excluded from the route planning, as they are no longer in the `active` state.

Finally, once arrived at the customer agent, the OD agent transitions to the `at_home` state and delivers another message to the selected customer agent. On receipt of the message, the customer agent transitions to the `CCS_complete` state. This state serves the purpose of keeping track of all outsourced customer agents until the commencement of regular company delivery. This allows the calculation of delivery savings by calculating a baseline cost for regular delivery which includes the customer agents in the `CCS_complete` state. At the execution of the `perform_deliveries` event, all customers in the `CCS_complete` state (along with those in the `awaiting_regular_delivery`) transition to the `latent` state.

4.6 Key performance indicators

A number of output variables are considered as indicative of the performance of the customer crowd-shipping system. The KPIs that are considered in this model primarily include the delivery cost, the company savings, and the customer waiting time. The delivery cost comprises the delivery vehicle cost and the cost of incentives paid, while the company savings captures the monetary savings made as a result of customer crowd-shipping.

4.6.1 Delivery cost

The first KPI considered relates to the cost of serving the customer agents in the system. This includes the delivery vehicle costs, as well as the cost of incentives paid to ODs.

Delivery vehicle costs are calculated after each set of regular deliveries. The total delivery cost is proportional to the distance travelled by the set of homogeneous vehicles. As mentioned in §4.3.3.2, the user-defined parameter `cost_per_distance` captures the cost per unit distance travelled by a delivery vehicle. It is assumed that the fixed costs of the delivery vehicle and the wage rate of the driver are included in the distance-based multiplier. As previously mentioned, the final delivery routes are determined by the `run_cvrp` function at each execution of the `fix_delivery_routes` event. The resultant distance captured in the objective function of the CVRP is multiplied by the `cost_per_distance` parameter to capture the total delivery vehicle cost. For each set of regular last-mile deliveries, the resultant total cost is captured and stored in the `delivery_vehicle_costs` dataset in the `main` agent environment. The summation of the data entries is therefore the total delivery vehicle cost for a given simulation run.

The cost of outsourcing deliveries is defined as the sum of the incentive paid to OD agents. Once an offer is accepted by an OD agent, the total incentive value λ_{ij} is temporarily stored in the agent's `accepted_incentive` variable. This value is also permanently captured in the `incentives_paid` dataset, defined in the `main` agent environment. From this dataset, the total incentives paid as well as the mean value of incentives paid may be determined.

4.6.2 Company delivery savings

The company delivery savings that result from crowd-shipping is, by definition, the difference in delivery cost between serving a set of customers with and without the use of customer crowd-shipping. Accordingly, let the baseline cost (*i.e.* the cost of last-mile deliveries without the use of customer crowd-shipping) be the cost of serving all customers in the system by means of the dedicated sets of deliveries as described in §4.3. The true delivery cost, however, considers the incentives paid to all OD agents that served customer agents, as well as the cost of serving the remaining customer agents with delivery vehicle agents. Thus, the company savings may be calculated at each delivery instance by subtracting the baseline delivery cost from the true delivery cost for that instance.

As described in §4.5.4, once a customer agent is served by an OD, they transition to the `CCS_complete` state. This state keeps track of the outsourced customers until the next set of regular deliveries are performed. Then, for a given execution of the `fix_delivery_routes`, the baseline cost is calculated as the cost of performing company delivery to the customer agents in the `CCS_complete` state as well as those currently in the `awaiting_regular_delivery` state. This baseline cost is captured in the `baseline_delivery_cost` dataset for each set of deliveries. The actual delivery cost, on the other hand, is calculated as the sum of incentives paid since the previous set of regular deliveries (*i.e.* for having outsourced the customer agents in the `CCS_complete` state) and the cost of performing regular deliveries to the customer agents in the `awaiting_regular_delivery` state. The actual delivery cost is captured in the `actual_delivery_cost` dataset for each set of deliveries. The company savings may then be calculated both as a monetary savings amount, and as a percentage delivery savings. The monetary savings value is calculated as the difference between the actual delivery cost and the baseline delivery cost for each set of deliveries and stored in the `delivery_savings` dataset. The percentage delivery savings, on the other hand, is calculated at the termination of each simulation run. The percentage savings per simulation run is calculated as the percentage difference between the sum of all entries in the `baseline_delivery_cost` dataset and the sum of all entries in the `actual_delivery_cost` dataset. More formally, the percentage savings is calculated as

$$\texttt{percentage\_savings} = \frac{\sum \texttt{baseline\_delivery\_cost} - \sum \texttt{actual\_delivery\_cost}}{\sum \texttt{baseline\_delivery\_cost}},$$

with the variable `percentage_savings` capturing the percentage savings per simulation run.

4.6.3 Customer waiting time

A major concern for last-mile delivery operations is the customer waiting time. To capture the customer waiting time, each customer agent has a variable `order_time`, which records the time at which they transition from the `latent` state into the `active` state. Once the customer agent is served, either by a delivery vehicle or by an OD agent, the service time is recorded. The working hours between the service time and the agent's `order_time` variable is captured each time a customer agent is served. The waiting time values are subsequently stored in the `waiting_time` dataset, which is defined in the `main` agent environment.

As a retailer would not be able to deliver during non-working hours, as defined in §4.2.3, the after-working hours are excluded from the waiting time. For example, if a customer agent places an order at 21:00 in the evening and is served at 09:00 on the next morning, only the time period from 08:00–09:00 is considered as waiting time.

4.7 The graphical user interface

A graphical user interface (GUI) is developed to enable the model user to initiate and interpret a simulation run. The GUI allows the user to change a number of interactive sliders, radio buttons, and textboxes that define the input parameters of a simulation run. Furthermore, it allows for the real-time interpretation of the model by translating the important elements into a display that is intuitive and informative. The following section describes the configuration screen as well as the real-time visualisations developed in the model.

4.7.1 Configuration screen

The configuration screen is displayed to the model user before the execution of a simulation run. The screen is developed in the `simulation` environment within the ANYLOGIC software and allows for the adjustment of a number of input parameters. When the simulation is initiated, the configuration screen is displayed with all sliders and textboxes on sensible, default values. These can then be adjusted by the model user within the sensible limits set out in the model. The configuration screen is shown in Figure 4.12 and the inputs displayed are discussed subsequently.

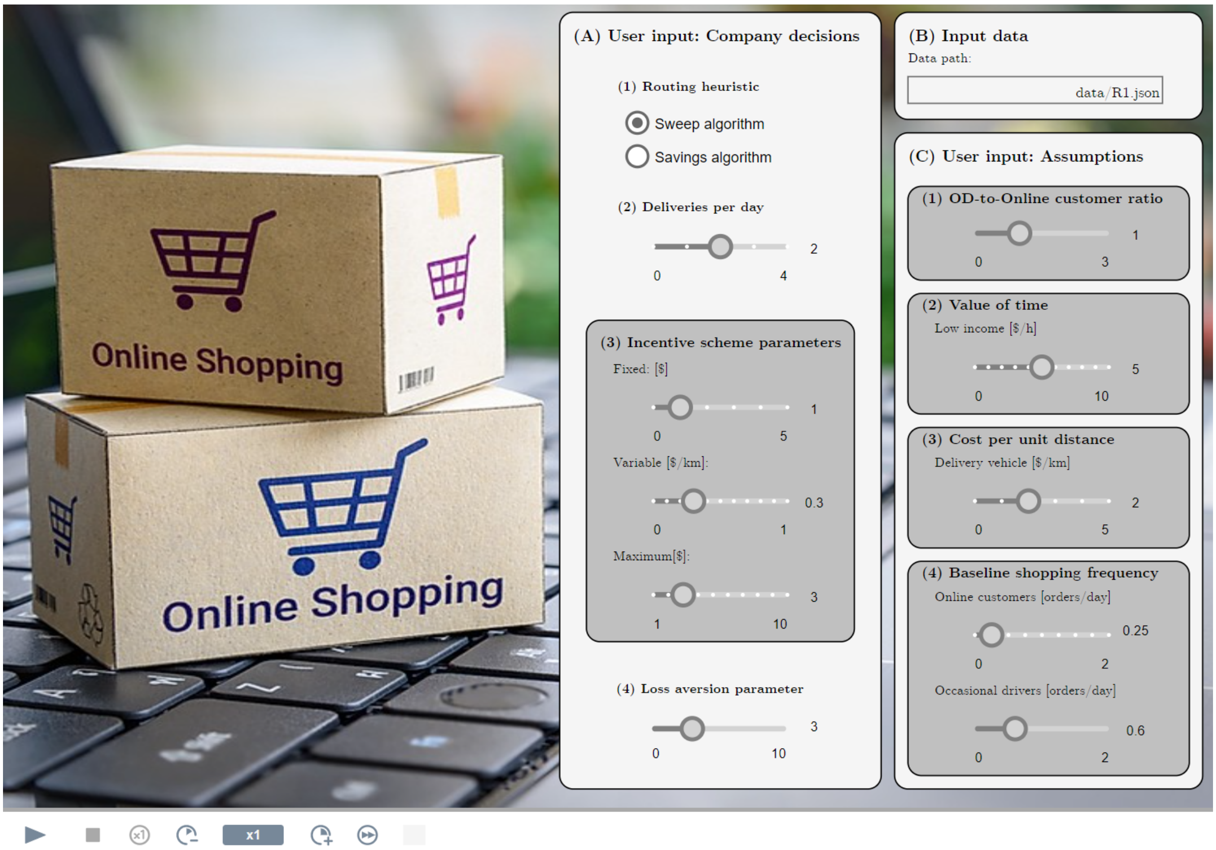


FIGURE 4.12: *The graphical user interface depicting the input parameters to the model that may be altered by the model user.*

4.7.1.1 User input: Company decisions

This collection of inputs, indicated in (A), represents the parameters that the retailer may have control over during the operation of a customer crowd-shipping initiative. The first input considered relates to the CVRP solution methodology employed in the `run_cvrp` function throughout the model execution. The selection of solution methodology relates to the routing heuristics discussed in §4.3.3.2, namely the savings and sweep algorithms. Although the selection of solution methodologies are limited to the savings and sweep heuristics, the modular nature of the proposed model allows for additional methodologies to be implemented.

Furthermore, the model user may set the frequency of the dedicated set of deliveries. This frequency is captured in the `deliveries_per_day` parameter and is limited to a value between 1 and 4. As discussed in §4.3.2, the deliveries occur according to a regular delivery schedule based on this parameter value.

Furthermore, the model user may change the parameters of the incentive scheme, as discussed in §4.5.2, which include the fixed, variable, and maximum incentive values. The fixed element of the incentive may take on values ranging from \$0.00 to \$5.00, and is set to a default value of \$1.00. Similarly, the variable rate may be adjusted to values between \$0.00 and \$1.00 per kilometre, with a default value of \$0.30 per kilometre. Finally, the maximum incentive may be adjusted to values between \$1.00 and \$10.00, with a default value of \$3.00. Once defined by the user, these incentive scheme parameters are captured in the `fixed_incentive`, `variable_rate`, and `max_incentive` parameters, respectively.

Finally, the loss aversion parameter, denoted by β, may be adjusted by the model user to influence the approach to selecting candidate orders for crowd-shipping. This parameter, along with the fixed incentive parameter and the cost-to-serve value of a customer agent, governs whether or not the customer agent is a candidate for crowd-shipping, as described in §4.5.3. The parameter is allowed to take on values in the range of 0–10 and is captured in the `loss_aversion` parameter. Accordingly, the user may vary the outsourcing approach from very liberal (*i.e.* by selecting a low value of the `loss_aversion` parameter) to more conservative (*i.e.* by selecting a high value of the `loss_aversion` parameter).

4.7.1.2 Input data

The input data shown at (B) allows for the CVRP data discussed in §4.2.1 to be extracted. The data path textbox captures the path to the `.json` file containing the specific CVRP benchmark problem set, and is stored in the `data_path` parameter. This dataset defines the locations and demands of the `customer` agent population, the location of the `depot` agent, as well as the capacity of the delivery vehicles, which is subsequently captured in the `vehicle_capacity` parameter. The model user may direct the data path to a company-specific dataset, however, which should include the aforementioned features of the system.

4.7.1.3 User input: Assumptions

The more complex set of parameters includes a number of assumptions made about the customer crowd-shipping system. This includes, as shown at (C), assumptions about the maturity of the customer crowd-shipping system, the OD population's value of time, the regular delivery vehicle cost, as well as the shopping frequencies of online customers and ODs. These values may vary significantly across different problem instances of the model.

The first assumption input parameter considered is the ratio between ODs and online customers. This parameter governs the number of OD agents compared to the number of customer agents within the system. As mentioned in §4.4, this input variable is captured in the `OD_customer_ratio` parameter, which is allowed to take on values in the range of 0–3. The lower end of this range (*i.e.* from 0.0–0.5) may represent a customer crowd-shipping initiative in its roll-out phase. As the system matures, it may be represented by a higher value of the `OD_customer_ratio` parameter. Finally, cases where the `OD_customer_ratio` parameter value is greater than 1.5, may represent a system that has reached full maturity and high levels of popularity.

The next assumption made is with respect to the mean value of time for an individual in the modelled low income bracket. As mentioned in §4.4.2.3, the value of time for all OD agents are calculated at the initialisation of a simulation run based on the mean value of time of the low income class. This input assumption is captured in the `minimum_VOT` parameter, which may take on values between \$2.00 per hour and \$10.00 per hour, and is set to a default value of \$5.00. This range emulates a conservative approach to the values of time found in literature, as described in §3.4.2.

Furthermore, the assumed cost per unit distance travelled by a delivery vehicle is set as an input parameter. As mentioned in §4.6.1, it assumed that the entire operational cost of a delivery vehicle may be estimated using a distance-based cost parameter. This may be unique for different implementations of the model and depends on the type of delivery vehicle used. A such, it is left to the model user to set the value, which is captured in the `cost_per_distance` parameter. This parameter may take on values between $1.00 and $4.00 per kilometre, with a default value of $2.00 per kilometre.

Finally, the mean order rate of customer agents and the shopping rate of OD agents are captured by the user as input. Both of these rates are governed by an exponential distribution and the user may define the mean values of these distributions. The customer agent ordering rate varies throughout the day, as discussed in §4.3.1. The user defines the mean value of the base order rate, which is captured in the `base_order_rate` parameter. Similarly, the base shopping frequency of an OD, as discussed in §4.4.1, is defined by the user and captured in the `base_OD_rate` parameter.

4.7.2 Visualisation

In addition to the configuration screen, the GUI involves tracking the state of the system as the simulation run is executed. The details of the run may be communicated visually to the user by means of animation and real-time plots. These visualisation tools are developed in the `main` agent environment within the AnyLogic software environment.

4.7.2.1 Animation of a simulation run

The various agent classes are represented by different icons to clearly depict their movement and behaviours. Furthermore, the state of an individual agent at a given moment in time may be identified by the use of colour. The icons and state identifiers for the various agent populations are detailed in Table 4.5.

Table 4.5: *The agent icons as used in the animation of a simulation run.*

Agent type	Icon	State	State identifier
Customer		`latent`	None
		`unassigned, sorting_substate`	●
		`regular_delivery_candidate`	●
		`CCS_candidate`	●
		`awaiting_regular_delivery`	●
		`awaiting_OD_delivery`	●
		`CCS_complete`	●
OD		`at_home, shopping`	None
		`delivering_order`	■
Delivery vehicle		`at_depot, active`	None
Depot		None	None

The `customer` agent population is displayed using the `house` icon. Once an order is placed, a circle appears above the top-left corner of the associated `house` icon, indicating that the state transition from `latent` to `active` has occurred. The colour of the circle graphically depicts the state of the customer agent as it transitions through the statechart. The `OD` agent population, on the other hand, are depicted by the `person` icon. To identify an instance of customer crowd-shipping, the icon colour changes to green once an offer has been accepted and the OD agent transitions to the `delivering_order` state. This correlates with the change in colour by the customer agent in the `awaiting_OD_delivery` state. The `depot` agent and the `delivery_vehicle` agent populations are depicted by the `warehouse` and `truck` icons, respectively, and require no state identifiers.

A screenshot of an example simulation run is shown in Figure 4.13. It depicts the `depot` agent with customer agents distributed around it. At this point in the simulation run, a number of customer agents in the `awaiting_regular_delivery` state are being served by delivery vehicle agents in the `active` state. The remainder of the delivery vehicles are stationary in the `at_depot` state. Furthermore, three customer agents have placed orders after the delivery route has been fixed. Two of these are in the `regular_delivery_candidate` state, while one is potentially subject to customer crowd-shipping and in the `CCS_candidate` state. Additionally, one customer agent is in the `awaiting_OD_delivery` state and is being served by an OD agent in the `delivering_order` state. The remainder of the OD agents are either in the `at_home` or `shopping` state. Finally, two customer agents have, during the period since the previous set of scheduled deliveries, been served by OD agents and are in the `CCS_complete` state. Customer agents are rarely observed in the `unassigned` or `sorting_substate` states, as these state identifiers are mainly used for verification purposes.

FIGURE 4.13: *A screenshot of the simulation screen during an example simulation run.*

4.7.2.2 Real-time plots of a simulation run

In addition to observing the behaviour and movement of agents, as depicted in Figure 4.13, the model user may observe a number of variables as they change over time. Certain key variables are plotted in real-time during the execution of a simulation run. A screenshot of these plots are shown in Figure 4.14.

FIGURE 4.14: *A screenshot of the real-time plots of variables during a simulation run.*

The first parameter displayed during a simulation run execution is the distribution of the `base_value_of_time` parameter for all OD agents. This is displayed as a histogram, as shown at (E), to provide insight into the value of time across different income brackets.

The remainder of variables that are displayed relate to the KPIs described in §4.6. First, the cumulative delivery costs over time are displayed, as indicated at (D), while distinguishing between the delivery vehicle costs and the incentives paid. The first plot shows the cumulative costs plotted over time, while the second plot provides a stacked version of the same data. An additional plot is shown at (G) to depict the distribution of the value of incentives paid. Furthermore, the distribution of the waiting time for all served customer agents are displayed as a histogram at (H), to provide insight into the effectiveness of the delivery system. Finally, a distribution of the company delivery savings for each delivery set is displayed as a histogram at (F), giving insight into the profitability of the customer crowd-shipping implementation for the given simulation run.

4.8 Chapter summary

In this chapter, the development of the proposed agent-based model for customer crowd-shipping was described. The modelling environment, including the ANYLOGIC simulation modelling environment, PYTHON and the PYPELINE custom library were elaborated upon in §4.1. This was followed by a general description of the model in §4.2, including a number of assumptions and limitations of the model. The first phase of the model development was described in §4.3, detailing the modelled last-mile delivery system of a traditional e-commerce retailer. In this section the CVRP was formulated and the heuristic solution methodologies were described. In §4.4, the second phase of model development was outlined with reference to the behaviour of ODs and their associated value of time. Finally, in §4.5, an integrated approach to customer crowd-shipping was proposed. This included an incentive scheme, a approach for identifying customers to propose as candidates for crowd-shipping, as well as a method for modelling OD decision making. The KPIs of the model were detailed in §4.6. Finally, the GUI developed was described in §4.7, both with respect to the initiation and interpretation of a simulation run.

CHAPTER 5

Model verification

Contents

In this chapter, verification methods are applied to the proposed simulation model, as described in Chapter 4. This includes the verification of the constituent elements of the model, as well as the overall functionality thereof. In §5.1, the necessity of verification, as well as a number of verification techniques are discussed. Thereafter, the application of these techniques to the proposed model is documented, confirming the correct functionality of the model. A bottom-up methodology is followed to adequately verify the model. In particular, the constituent elements of the model are sequentially verified in order of increasing complexity and dependency. Accordingly, the first verification tests involve a number of elements discussed in the first phase of model development. This includes the verification of the customer order frequency in §5.2, as well as the implementation of vehicle routing and last-mile deliveries in §5.3. Once the functionality of the vehicle routing heuristics are verified, the implementation of the cost-to-serve algorithm is verified in §5.4. Focus is then shifted to the behaviour of ODs in §5.5, verifying a number of elements described in the second and third phase of model development. Finally, as a culmination of all the constituent elements, the correct calculation of the delivery savings is verified in §5.6.

5.1 Model verification

A vital step in a simulation study, as described in §3.2.2, involves determining whether or not the simulation model is an adequate representation of the real-world system being studied. Verification is the process of debugging a computer program to ensure that it performs as intended, thereby confirming that the assumptions made about the system is reflected in the software program. Law [126] proposes a number of verification techniques which may be applied to the development of the proposed customer crowd-shipping model.

First, Law [126] proposes that the constituent elements of a simulation model should be debugged during the process of development. Rather than constructing the entire model and attempting to debug it post-*hoc*, the model complexities should be developed and debugged concurrently. This iterative approach was applied throughout the development of the customer crowd-shipping model by continually compiling, executing, and debugging the simulation model. Only once the model was sufficiently verified, a new layer of complexity was added. Moreover, such an approach to verification is well suited to the phased approach that was undertaken for model development. It is ensured that each phase is adequately verified and functioning as expected before considering the additional complexities of the subsequent phase. This technique may be applied in combination with the *trace* technique, whereby the state of the simulated systems are displayed throughout the execution of the simulation run. Additionally, the use of animation is proposed, as the resulting output may be clearly demonstrated with limited interpretation required. Finally, it is proposed that the model may be run under simplifying conditions, as the expected output may be trivial or easily derived.

In the verification tests to follow, a combination of the aforementioned techniques are applied. The AnyLogic Software Suite is equipped with a graphics terminal, interactive run controller, and a console for trace purposes. This allows for the verification tests to be displayed graphically as the simulation executes. It is furthermore complemented by the console output, verifying the state of individual agents and their associated parameters, variables, and other modelling elements. The datasets used for verification purposes are limited to 25 online customers in order to simplify the conditions and ease the tracking of individual agents.

In addition to verification within AnyLogic, the Pypeline custom library is utilised in analysing the output of the verification tests. Given that many of the computational problems, such as solving the CVRP and calculating cost-to-serve values, are executed using Python, the verification of these computations are performed using Python libraries. This is enabled by the generation of highly customisable visualisations using libraries such as matplotlib. Finally, by comparing the output generated in the Python environment with the resulting actions of agents in the AnyLogic environment, the functional integration of these environments (*i.e.* the operation of the PyCommunicator object) is additionally verified.

5.2 Customer order frequency

As described in §4.3.1, the rate at which customer agents place orders is governed by an exponential distribution with a mean rate of λ. This rate is dependent on the user-defined parameter `base_order_rate`, as well as the pre-defined schedule `daily_demand`. In this section, verification tests are conducted to ensure that these parameters are reflected adequately in the execution of simulation runs. First, the effect of the `base_order_rate` parameter value is verified. This is followed by verification of the influence of the `daily_demand` schedule.

5.2.1 Base order rate parameter

In order to ensure that the `base_order_rate` parameter value influences the rate of online orders adequately, the resultant order frequency is observed for varying values of the `base_order_rate` parameter. To this end, six experiments are conducted, each relating to a simulation configuration with a different value of the `base_order_rate` parameter. In particular, each experiment considers the average resultant order rate of 25 customer agents over a simulated timeframe of 25 days. In order to increase the level of confidence in the observed resultant order rate, each simulation configuration is run ten times with a random variable seed, resulting in ten independent replications. The average order frequency across the ten replications are then considered as an estimate for the particular configuration. For Experiments 1.1–1.6, the `base_order_rate` parameter is set to the values of {0.125, 0.25, 0.50, 1.00, 1.50, 2.00}, respectively. The `deliveries_per_day` parameter is set to a value of 2, and there are no ODs in the system with the `OD_customer_ratio` set to a value of 0. To model a constant order rate throughout the day, the daily demand rate factor, governed by the `daily_demand` schedule, assumes a value of 1 from 07:00–22:00, and 0 for all other times.

The results obtained from Experiments 1.1–1.6 are tabulated in Table 5.1. First, the expected order rate per customer is shown (*i.e.* the value of the `base_order_rate` parameter). Thereafter, the average total orders placed by the 25 customers agents per replication, as well as the resultant order rate per customer agent is shown. Finally, the percentage difference in the expected and resultant order rates are shown.

TABLE 5.1: *Experiments 1.1–1.6: Comparison of expected and resultant order rate for 25 customer agents over a simulated timeframe of 25 days.*

Experiment	Expected rate per customer [orders/day]	Average resultant total orders [orders]	Average resultant rate per customer [orders/day]	Percentage difference
1.1	0.125	76.30	0.12	2.34%
1.2	0.25	144.60	0.23	7.46%
1.3	0.50	292.80	0.46	6.30%
1.4	1.00	557.50	0.89	10.80%
1.5	1.50	671.20	1.07	28.41%
1.6	2.00	742.90	1.19	40.57%

It is clear that as the `base_order_rate` parameter value increases, there is an associated increase in the average total orders and, accordingly, the resultant order rate. There is, however, a discrepancy between the expected and resultant order rates, with the latter being less for all experiments. The degree to which these rates differ are captured in the percentage difference, which is shown to escalate with an increasing expected order rate. The primary findings of Table 5.1 are illustrated more clearly in Figure 5.1. In this figure, the average expected order rates are juxtaposed with the average resultant order rates for Experiments 1.1–1.6.

It is evident from Figure 5.1 that the difference between the expected and resultant order rates increases with an increasing expected order rate. The magnitude of the percentage difference regresses to more than 10% for expected order rates of greater than one order per day.

To provide more insight into the discrepancies shown, the distribution of order placement throughout the simulation runs is investigated. The average hourly order frequencies per simulation run are shown in Figure 5.2. More specifically, the average order frequencies for Experiments 1.1–1.6 are shown in Figures 5.2(a)–5.2(f), respectively.

The order distribution throughout the day changes from the expected uniform distribution, as observed in Experiments 1.1–1.4, to a bimodal distribution observed clearly in Experiment 1.5 and Experiment 1.6. As shown in Figure 4.2, when implementing the model with a `deliveries_per_day` parameter value of 2, the scheduled sets of deliveries are initialised at 11:00 and 17:00, respectively. In considering this, it may be observed from Figure 5.2(e) and Figure 5.2(f) that the highest influx of orders occur during the hours directly after the sets of

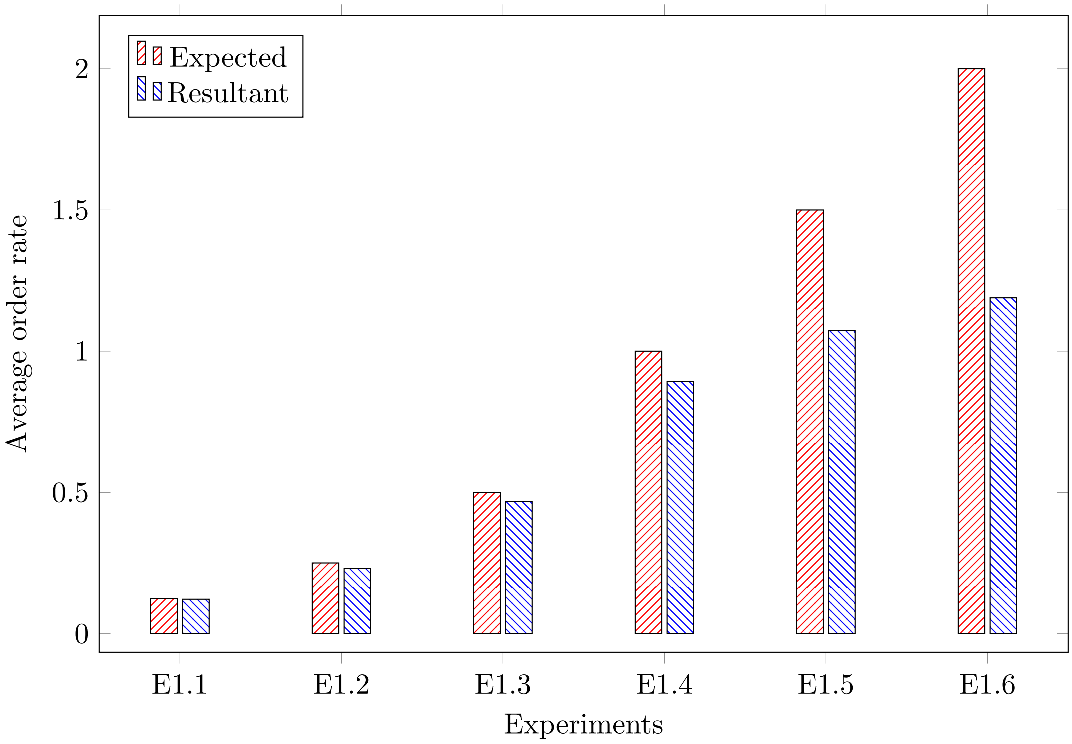
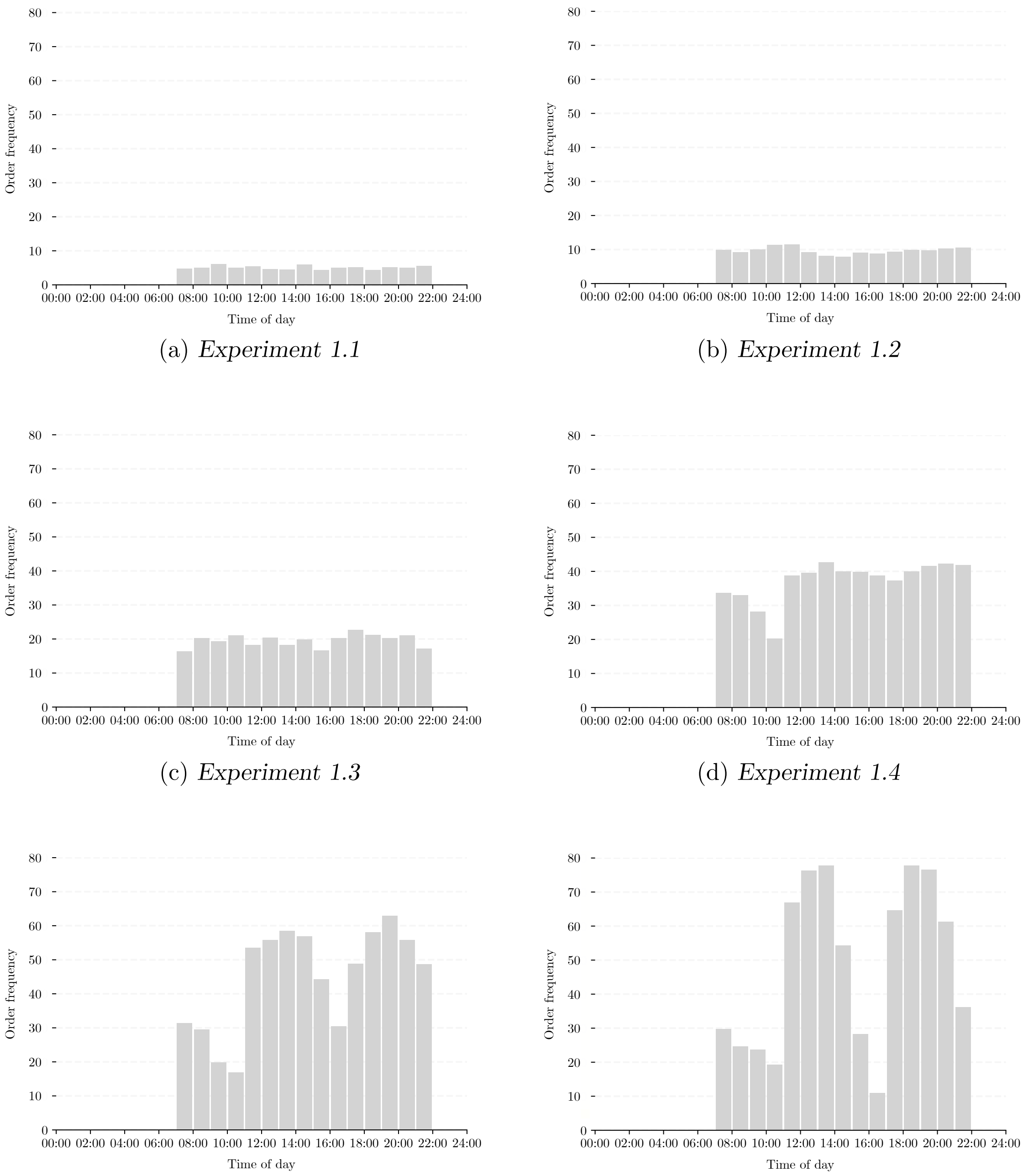


FIGURE 5.1: *The expected versus actual average order frequencies.*

deliveries are initiated for both Experiment 1.5 and Experiments 1.6. This indicates that the occurrence of a set of deliveries may have an influence on the manner in which customer agents place orders. In particular, it is found that the `base_order_rate` parameter value relative to the `deliveries_per_day` parameter value has an influence on the orders placed in the model. As the customer order rate approaches the delivery frequency, the expected order rate distribution is distorted. This is because customer agents may only place an order while in the `latent` state (*i.e.* customer agents may not place an additional order while waiting for the delivery of an existing order). As such, while customer agents wait for the delivery of an existing order, the occurrence of the `place_order` event has no influence on them. When the order frequency is too high, there are extended time periods during which all customer agents are in the `active` state and, as such, no additional orders can be placed. This effect may first be observed in Figure 5.2(d), where there is a slight decrease in the resultant order rate before the execution of the `perform_deliveries` event at 11:00. Similarly, in Figure 5.2(e) and Figure 5.2(f), declines in order frequencies are observed before the execution of the `perform_deliveries` events at 11:00 and 17:00, respectively. Once the set of deliveries are complete, all customer agents are in the `latent` state and the resultant order rate increases.

Furthermore, this observation explains the results from Table 5.1 and Figure 5.1. With the `base_order_rate` parameter value approaching the `deliveries_per_day` parameter value, there are increasingly long periods during which all customer agents are awaiting delivery and are unable to place orders. This distorts the total orders placed and results in an increasing percentage difference between the expected and resultant order rates. To ensure a percentage difference smaller than 15%, it is advised that the `base_order_rate` parameter is set to a value less than half of the `deliveries_per_day` parameter value. Once this condition is not met, the model does not behave as the user input would suggest.



(a) *Experiment 1.1* (b) *Experiment 1.2*

(c) *Experiment 1.3* (d) *Experiment 1.4*

(e) *Experiment 1.5* (f) *Experiment 1.6*

FIGURE 5.2: *The average order rate distribution throughout the day for Experiments 1.1–1.6.*

5.2.2 Verification of schedule influence

In verifying the influence of the **daily_demand** schedule on the resultant order rate, an additional verification experiment is conducted. This is achieved by adjusting the values of the schedule and analysing the resulting order distribution.

In Experiment 2, the demand factor values of the **daily_demand** schedule reflect the values described in §4.3.1 and tabulated in Table 4.3. This relates to a value of 0.5 during regular working hours (*i.e.* 07:00–17:00), a value of 1.2 for after-work hours (*i.e.* 17:00–22:00), and a value of 0 for the remainder of the day in which customers are unlikely to place orders. Furthermore, the value of the base_order_rate parameter is set to 0.5 orders per day, and the resultant order rate of 25 customer agents are observed over a simulated timeframe of 25 days. The deliveries_per_day parameter is set to a value of 2 deliveries per day, while the OD_customer_ratio parameter value is set to 0. To increase confidence in the result and negate the influence of stochasticity, ten replications of the experiment are conducted with a variable random seed. The average resultant order distribution for the ten replications is recorded, and is shown in Figure 5.3.

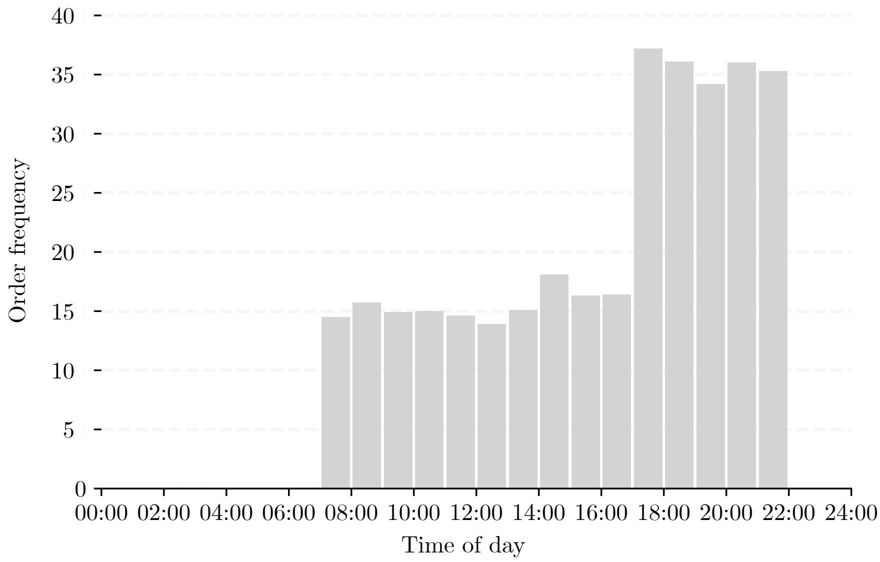


FIGURE 5.3: *The average order rate distribution throughout the day for Experiment 2.*

In Figure 5.3, there is a clear distinction between the average order rate from 07:00–17:00 and that of 17:00–22:00. The increase in rate during non-working hours reflects the values in the daily_demand schedule. Furthermore, the values of the daily_demand schedule are calibrated such that the average resultant order rate reflects the base_order_rate parameter value. In Experiment 2, the average resultant order rate across the ten replications was found to be 0.53, which closely reflects the expected value of 0.5.

5.3 Vehicle routing and last-mile delivery

In this section the correct implementation of the regular last-mile deliveries is verified. This is primarily focused on Phase I of the model development, as described in §4.3. First, it is verified that deliveries are performed as per the expected schedule and that company delivery costs are calculated correctly. Furthermore, it is verified that all appropriate customer agents are selected for delivery. The section is concluded by verifying that the routing heuristic employed solve the CVRP adequately, while the delivery vehicle agents serve all appropriate customer agents, as determined by the heuristic.

The locations of the customer agents in the verification tests conducted in this section are defined by a Solomon problem set [183], described in Appendix A. In particular, the uniform randomly distributed problem set, R1 for 25 customers, is utilised for the experiments described in this section. The locations of the customer agents and depot are illustrated graphically on a Cartesian plane in Figure 5.4. The simplifying conditions of a small number and sparse distribution of customer agents may improve the interpretability of the test results. The delivery vehicle capacity, however, is altered from that defined in the Solomon problem instance, and set to a value of 100. The reduced capacity enables clearer demonstration and verification of the routing and last-mile delivery.

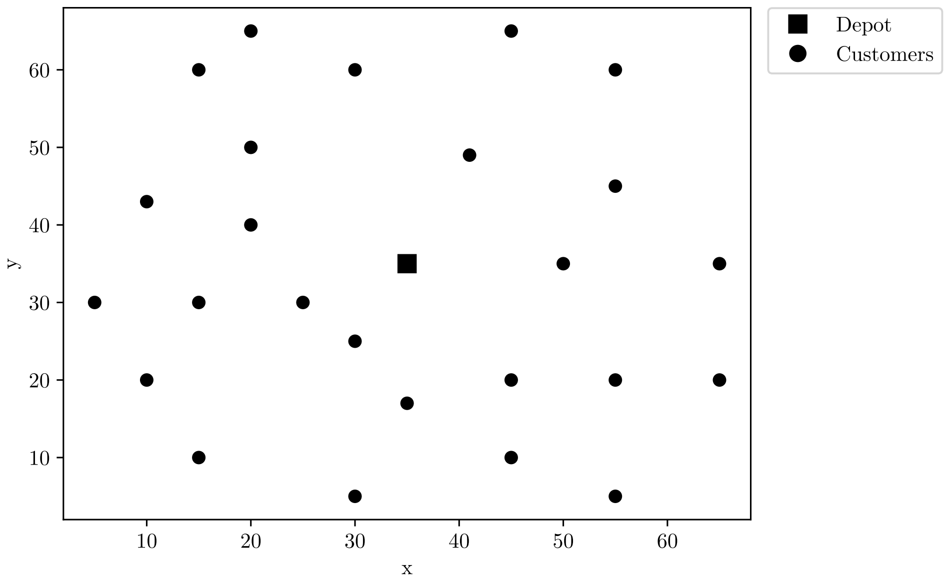


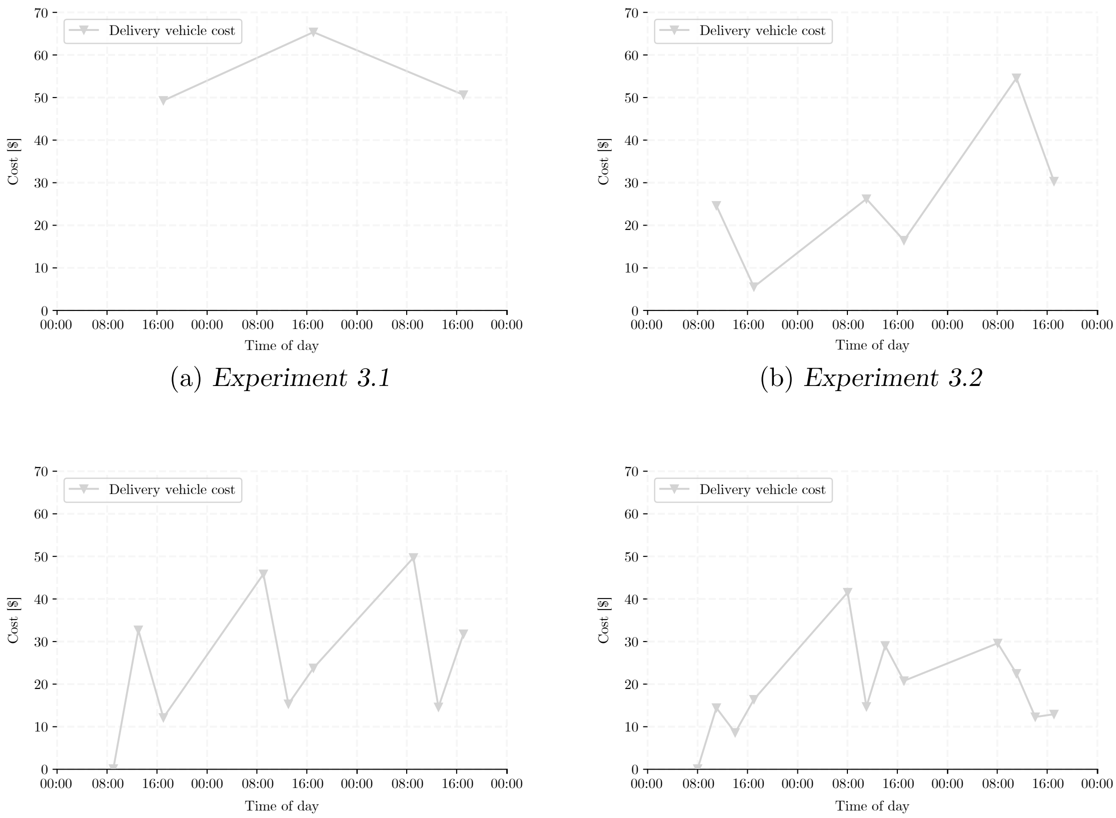
FIGURE 5.4: *The locations of customers and the depot of the R1 problem set for 25 customers from Solomon [183].*

5.3.1 Delivery schedule and cost

As described in §4.3.2, the model user has the option of adjusting the frequency at which the company performs deliveries to customers. Verification tests are performed to ensure that the frequency of deliveries reflects the user-defined value of the `deliveries_per_day` parameter. These tests are furthermore used to verify that the deliveries are performed as per the schedule illustrated in Figure 4.2. Finally, tests are conducted to ensure that the delivery cost is adequately influenced by the `cost_per_distance` parameter, as described in §4.7.1.

For Experiments 3.1–3.4, the `deliveries_per_day` parameter value is varied between 1 and 4 deliveries per day. By observing the `delivery_vehicle_costs` dataset over time, the frequency and schedule of deliveries may be verified. The parameters `base_order_rate` and `cost_per_distance` are set to values of 0.5 orders per day and $2.00 per kilometre, respectively, for all the relevant experiments. To observe the regular deliveries in isolation, no ODs are utilised in the system and the `OD_customer_ratio` parameter value is set to 0. For each experiment, the simulation is run for a simulated time of three days with a fixed seed for the random number generator, enabling reproducible runs. The `delivery_vehicle_costs` dataset is plotted over this time in Figure 5.5. In particular, the results of Experiment 3.1–3.4 are displayed individually in Figures 5.5(a)–5.5(d).

From the results shown in Figure 5.5, it can be seen that the sets of deliveries are executed at the expected frequency while the schedule proposed in Figure 4.2 is adhered to in all experiments. In particular, for Experiment 3.1 a single set of deliveries are executed at 17:00 on each of the



(a) *Experiment 3.1*

(b) *Experiment 3.2*

(c) *Experiment 3.3*

(d) *Experiment 3.4*

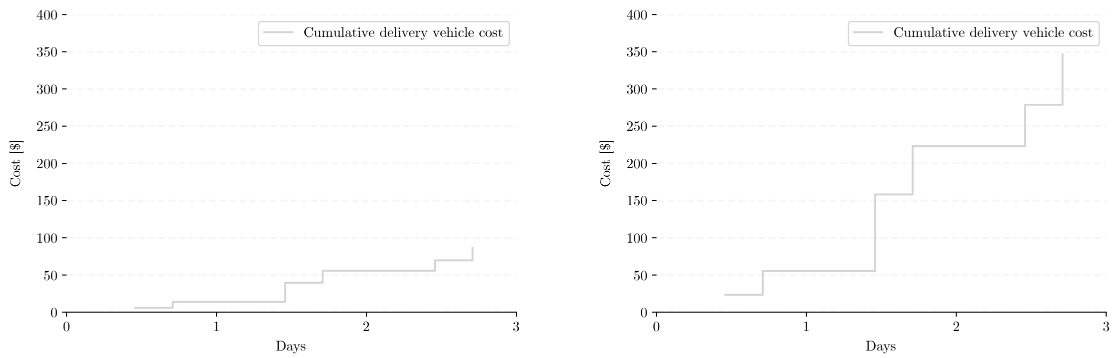
FIGURE 5.5: *The delivery costs over three days for various of delivery frequencies.*

three days. Similarly, for Experiment 3.2, two sets of deliveries are executed daily at 11:00 and at 17:00. Furthermore, for Experiment 3.3 three sets of deliveries are executed on each of the three days, at 09:00, at 13:00, and at 17:00, respectively. Finally, for the Experiment 3.4, it is evident that four sets of deliveries are executed at 08:00, 11:00, 14:00, and at 17:00, on each of the three days.

An additional observation is made in Figures 5.5(c) and 5.5(d), that the first set of deliveries performed in Experiments 3.3 and 3.4 both result in a cost of $0.00. At the higher delivery frequencies, observed in Experiments 3.3 and 3.4, the first set of daily deliveries is executed relatively early — at 09:00 and 08:00, respectively. Since a simulation run starts at 08:00, there are few (or no) customer agents awaiting delivery within the first few hours of the first day. For the subsequent days, however, customer agents may place orders during the evening, resulting in a non-zero delivery cost for the first set of deliveries of the day. In fact, the higher after-work order rate defined in the `daily_demand` schedule results in the highest delivery cost being associated to the day's first set of deliveries. The phenomenon on the first day of the simulation run is referred to as the simulation *warm-up* period, which is more pronounced at higher delivery frequencies. In order to avoid the effect of the warm-up period, the results of the first day of the simulation run are omitted from model analysis. The warm-up effect, however, seems to be inconsequential for simulation runs of an adequate time-duration.

The effect of the `cost_per_distance` parameter is verified in Experiments 4.1 and 4.2, during which the `cost_per_distance` parameter value is varied from $1.00 to $4.00 per kilometre. By observing the cumulative cost of deliveries over time, which is captured in the `delivery_vehicle_costs` dataset, the cost per unit distance is verified. The parameters `base_order_rate` and `deliveries_per_day` are set to the values of 0.5 orders per day and 2 deliveries per day, respectively, for both experiments. Finally, no ODs are utilised in the system, and the `OD_customer_ratio` parameter value is set to 0.

For Experiments 4.1 and 4.2, the simulation is run for a simulated duration of three days with a fixed seed for the random number generator, allowing for a direct comparison between runs. The cumulative value of the `delivery_vehicle_costs` dataset is plotted over this time period in Figure 5.6. The results of Experiments 4.1 and 4.2 are displayed individually in Figures 5.6(a) and 5.6(b), respectively.



(a) *Experiment 4.1* (b) *Experiment 4.2*

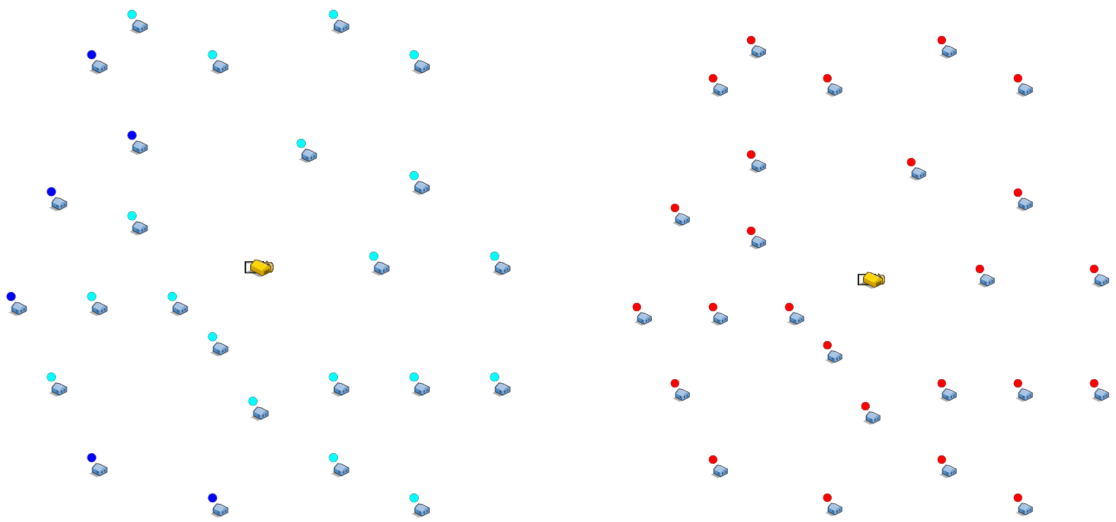
FIGURE 5.6: *The cumulative delivery costs over three days for various values of the cost per distance parameter.*

It can be deduced from Figure 5.6 that the deliveries were performed at the same times, serving the same customer agents in each run. The costs differ, however, with Experiment 4.1 resulting in a cumulative delivery vehicle cost of \$87.00, while the total delivery cost in Experiment 4.2 amounted to \$384.00. This verifies the influence of the `cost_per_distance` parameter.

5.3.2 Customer selection for delivery

As discussed in §4.5.3, all customer agents in the `active` state, regardless of their cost-to-serve value, are selected for regular deliveries prior to a set of scheduled deliveries. Specifically, 30 minutes prior to the commencement of the set of deliveries, the `fix_delivery_routes` event broadcasts a message to all customer agents in the `active` state. This message induces the transition of customer agents, both in the `CCS_candidate` and `regular_delivery_candidate` states, to the `awaiting_regular_delivery` state. This section describes the verification test conducted to verify that all appropriate customers are selected, receive the message, and transition as expected, to be served subsequently.

Experiment 5 considers the scenario where all customer agents in the system have placed an order and are currently in the `active` state. To purposefully induce this scenario, the `base_order_rate` parameter is set to an extreme value of 6 orders per day, while the value of the `deliveries_per_day` parameter is set to 1 delivery per day. Thus, it is expected that all customer agents would have placed orders before the delivery event at 17:00. Furthermore, to ignore the effect of customer crowd-shipping, no ODs are included in the system, with the `OD_customer_ratio` set to a value of 0. The simulation is executed with a fixed seed for the random number generator, until the completion of the first set of deliveries. In Figure 5.7, two screenshots of the simulation screen are shown, illustrating the state of customer agents before and after the occurrence of the `fix_delivery_routes` event. In Figure 5.7(a), the state of the simulation is shown at 16:29, with some customer agents in the `CCS_candidate` state and the remainder in the `regular_delivery_candidate` state. In Figure 5.7(b), the customer agents are shown at 16:30, directly after the `fix_delivery_routes` event, having transitioned to the `awaiting_regular_delivery` state.



(a) *Customers agents before routes are fixed* (b) *Customers agents after routes are fixed*

FIGURE 5.7: *Customer agents awaiting delivery before and after fixing the delivery routes.*

It is verified that the message broadcasted by the `fix_delivery_routes` event induces the correct transitions between states. All customer agents that are awaiting delivery, irrespective of their cost-to-serve value, transition to the `awaiting_regular_delivery` state. Thus, it is confirmed that if a customer agent has not yet been served by an OD agent, they will be added to the scheduled set of regular deliveries.

5.3.3 CVRP verification

In a continuation of Experiment 5, it is verified that the CVRP is formulated and solved as expected. As described in §4.3.3, the `run_cvrp` PYTHON function formulates and solves the CVRP for all customer agents in the `awaiting_regular_delivery` state. The set of customer agents that have been selected for regular delivery, as shown in Figure 5.7(b), is considered for the formulation of the CVRP as described in §4.3.3.1. Furthermore, the model user's choice of solution methodology, as described in §4.7.1.1, is investigated. This includes the verification of both the sweep and savings algorithms, as described in §4.3.3.2, in providing feasible solutions to the CVRP.

5.3.3.1 Verification of the sweep algorithm

The first phase of the sweep algorithm, as described by Algorithm 4.2, involves the formation of capacitated clusters. To ensure that the clusters are formed as expected, the output of the algorithm's first phase during the simulation is plotted graphically in Figure 5.8. Each cluster of customers shown is associated to a specific vehicle and, as such, the cumulative demand of a cluster should be less than the vehicle capacity. The cumulative demand associated with each vehicle according to the algorithmic output is tabulated in Table 5.2, where the maximum vehicle capacity is 100.

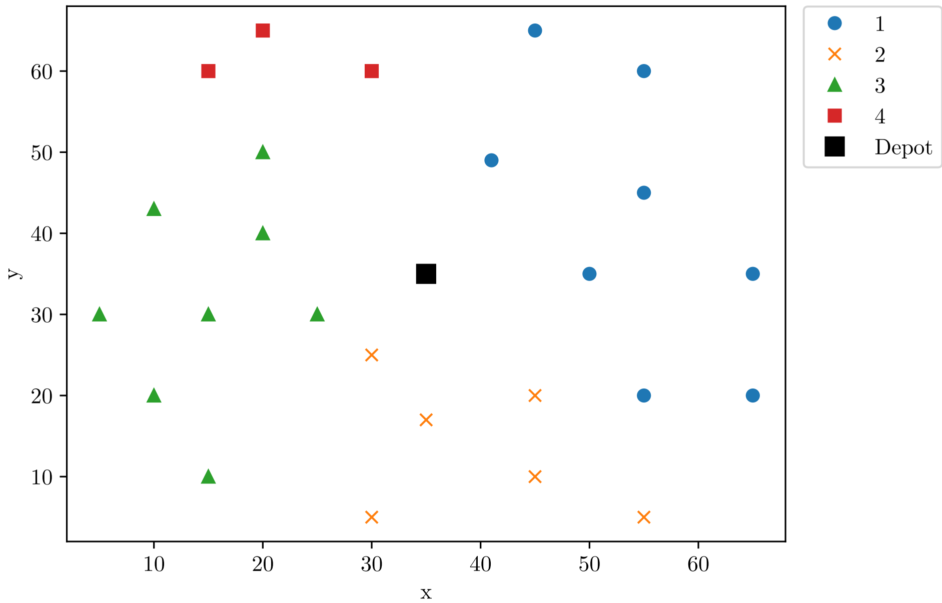


FIGURE 5.8: *Clustered customers according to the sweep algorithm.*

TABLE 5.2: *A verification of cumulative demand per vehicle for the sweep algorithm.*

Vehicle	Cumulative demand
1	95
2	96
3	96
4	45

The clusters are formed as expected, adding customers in a clockwise fashion starting directly above the depot. Additionally, the capacity limitations of all vehicles are adhered to, as governed by Constraint Sets 4.6 and 4.7. The first three delivery vehicle agents used are filled to the point where their capacity constraints limit additional customers. For the final delivery vehicle agent utilised, on the other hand, all customer agents are served before its capacity is reached.

In the second phase of the sweep algorithm, the clusters are treated as individual TSPs. For the clusters illustrated in Figure 5.8, the resultant final routes, as determined by the algorithm during the simulation run, are illustrated graphically in Figure 5.9.

From this result, it is visually verified that the remainder of the constraints, as described in §4.3.3.1, are satisfied through the use of the sweep algorithm. First, it is shown that all customer agents are served by a delivery vehicle agent, that arrives at and departs from the customer agent's location, in adherence to Constraint Sets 4.2 and 4.4. Furthermore, it is shown that Constraint Sets 4.3 and 4.5 are adhered to, given that the routes of all delivery vehicle agents originate from and terminate at the depot. The delivery vehicle agents that are not necessary

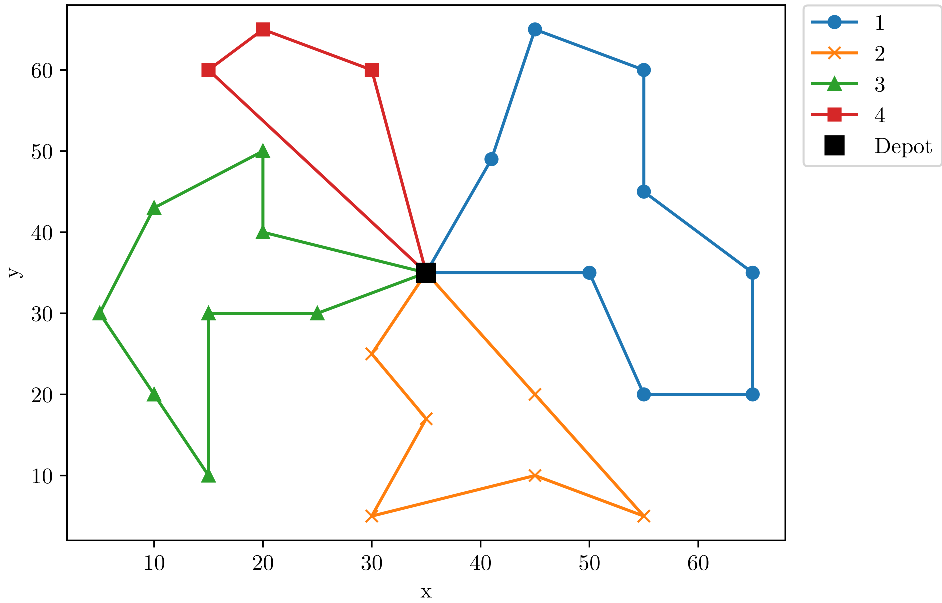


FIGURE 5.9: *The final routes for the sweep algorithm verification.*

in serving the cumulative customer demand may remain at the depot, by setting the decision variable $x_{0,n+1,k} = 1$ for each delivery vehicle k that is not utilised. Finally, it is visually verified that no subtours are formed, in final verification of adherence to Constraint Sets 4.6 and 4.7.

5.3.3.2 Verification of the savings algorithm

For the implementation of the savings algorithm, the resultant routes are formed constructively by making the best feasible merges. To verify that the merges are indeed feasible and that all constraints are adhered to, the savings algorithm is applied to the scenario depicted in Figure 5.7(b). The final set of routes, as determined by the savings algorithm, is shown in Figure 5.10.

The routes illustrated in Figure 5.10 allows for visual verification that the solution of the savings algorithm adheres to the CVRP constraints described in §4.3.3.1. First, each customer agent is served by a delivery vehicle agent that arrives at and departs from the customer agent's location, in adherence to Constraint Sets 4.2 and 4.4. Furthermore, the routes of all delivery vehicle agents originate from and terminate at the depot, in adherence to Constraint Sets 4.3 and 4.5. Finally, the cumulative demand of each route is tabulated in Table 5.3, whereas it is visually verified that no substours are formed. As such, adherence to Constraint Sets 4.6 and 4.7 are verified.

TABLE 5.3: *A verification of cumulative demand per vehicle for the savings algorithm.*

Vehicle	Cumulative demand
1	64
2	91
3	94
4	83

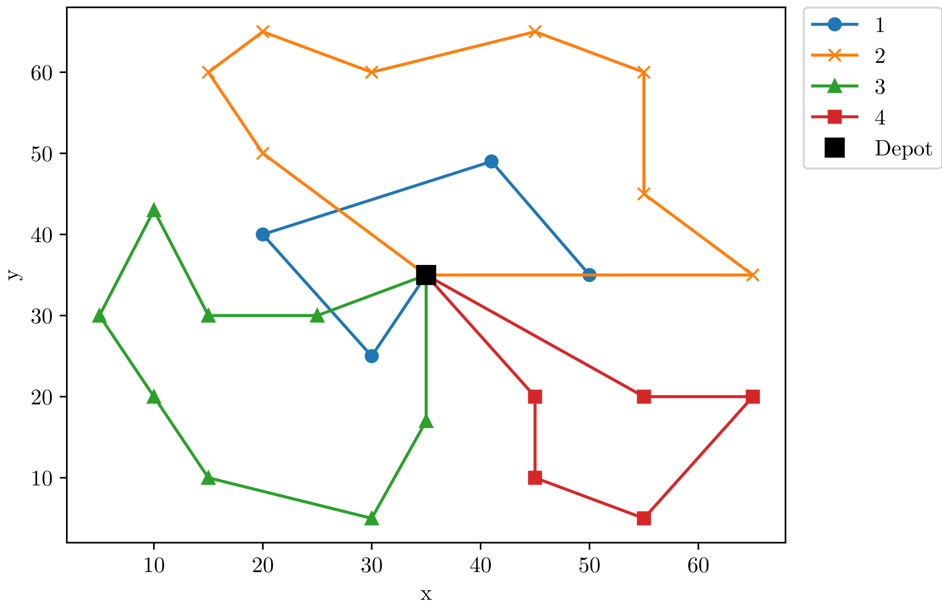


FIGURE 5.10: *The final routes for the savings algorithm verification.*

In the final part of Experiment 5, it is verified that the CVRP solution is correctly translated to the agent-based model. Specifically, it is verified that all customer agents are assigned to and served by the appropriate delivery vehicle agents. In verifying this aspect, the solution of the sweep algorithm presented in Figure 5.9 is translated to the simulation model, with each optimised TSP being assigned to a delivery vehicle agent. The progress of the delivery vehicle agents on a delivery run are then tracked and illustrated to ensure appropriate behaviour. A screenshot of the simulation run at 17:01 captures the commencement of deliveries and is provided in Figure 5.11. It illustrates each delivery vehicle agent and their respective route to serve their associated customer agents.

To ensure that the delivery vehicle agents serve all customer agents on their route, their delivery progress is tracked. As described in §4.3.4, as soon as a customer agent is served, a message transmitted from the delivery vehicle agent results in the transition of the customer agent from the ● `awaiting_regular_delivery` state to the `latent` state. This is illustrated graphically in Figure 5.12, wherein customer agents that are yet to be served are shown with red state indicators, verifying their ● `awaiting_regular_delivery` state. Conversely, customer agents that have been served are shown to be in the `latent` state, having no state indicator.

Experiment 5 therefore successfully verifies multiple aspects of the company delivery. First, it is successfully verified that all appropriate customer agents are selected for delivery. Furthermore, it is verified that the CVRP is appropriately formulated and solved for the selected customer agents, according to the model user's selection of solution methodology. Moreover, the implementation of the sweep and savings algorithms are thoroughly verified by ensuring their respective solutions adhere to the constraints detailed in §4.3.3.1. Finally, it is verified that this solution of the CVRP is correctly translated to the agent-based model and that all customer agents are subsequently served by the appropriate delivery vehicle agent along the appropriate route as per the CVRP solution.

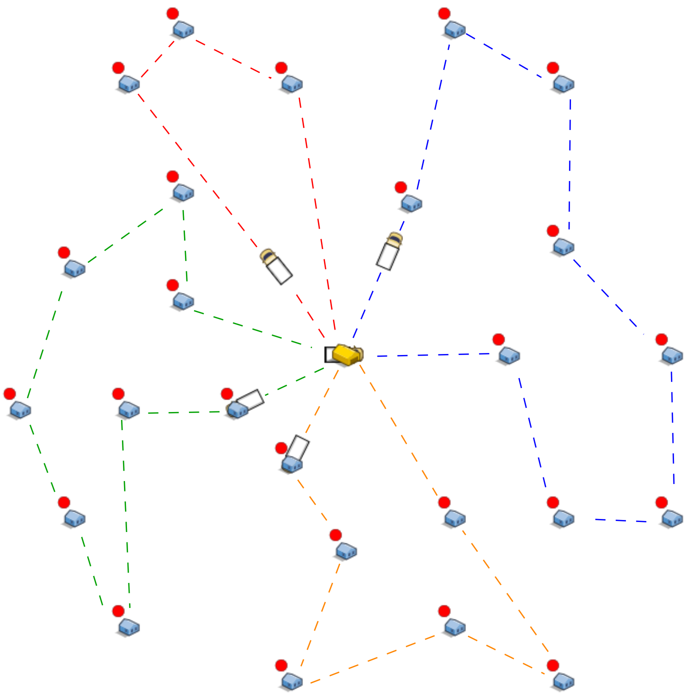


FIGURE 5.11: *The commencement of the set of last-mile deliveries according to the sweep algorithm solution.*

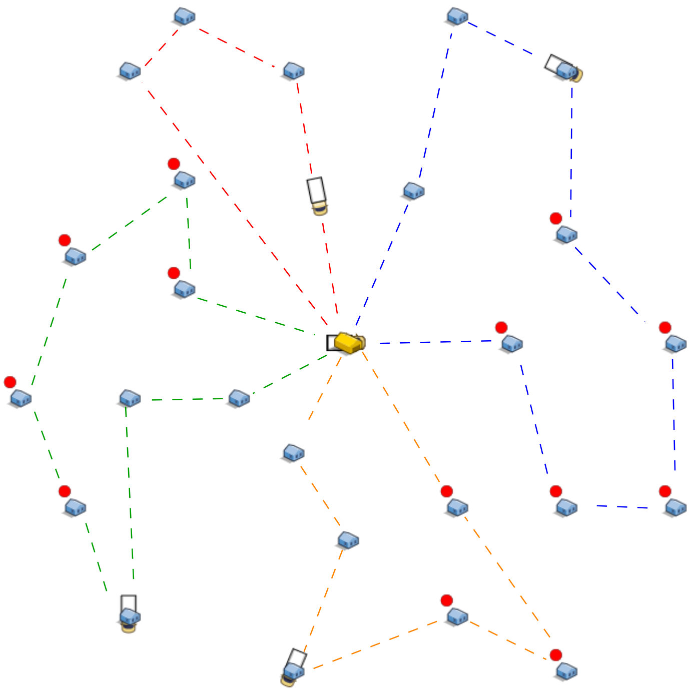


FIGURE 5.12: *The progress during a set of last-mile deliveries according to the sweep algorithm solution.*

5.4 Cost to serve

As described in §4.5.1, each customer agent has a `cost_to_serve` variable that captures the cost that would be incurred for serving the customer agent with a delivery vehicle agent at that moment in time. It is updated each time a new order is placed (*i.e.* each time the delivery route changes) by means of the `calculate_cost_to_serve` function. The cost-to-serve algorithm utilised in this function is described in §4.5.1. In the following section, the workings of Algorithm 4.3 is verified, and the potential results of the algorithm are discussed. Finally, it is verified that the `cost_to_serve` variable of all customer agents are updated appropriately in the model during a simulation run.

For the purpose of the verification test in this section, a test dataset containing information of nine customer agents are utilised. In Table 5.4, the x-coordinates, y-coordinates, and demand of each customer agent are tabulated.

TABLE 5.4: *A verification dataset containing the locations and demand of nine customers and a depot.*

Customer (i)	x-coordinate	y-coordinate	Demand
Depot	40	50	-
1	40	69	20
2	45	70	30
3	60	0	40
4	40	30	50
5	25	50	10
6	30	55	20
7	20	75	5
8	22.5	80	10
9	25	85	20

5.4.1 Verification of cost-to-serve algorithm

In verifying the workings of Algorithm 4.3, as well as for the demonstration of a number of interesting cases, the cost-to-serve values for the set of nine customers in Table 5.4 are calculated in Experiment 6. For demonstration purposes, the sweep algorithm is utilised with delivery vehicles with a capacity of 70. Finally, in order to simplify the interpretation of the cost-to-serve calculation, the `cost_per_distance` parameter is set to a value of \$1.00 per kilometre.

Initially, routing is performed for the entire set of customer agents, as illustrated in Figure 5.13, resulting in a maximum routing cost of R_{max} = \$29.52. Thereafter, the separate routing cost R_i is calculated by performing the routing algorithm for all customer agents except customer agent i, as illustrated in Figure 5.14. In particular, Figures 5.14(a)–5.14(i) illustrate the resultant routes calculated by the sweep algorithm while omitting one customer agent for each instance. Finally, the cost-to-serve value c_i of each customer agent i is calculated by subtracting R_i from R_{max}. The resultant R_i and c_i values of the customer agents in the verification dataset are captured in Table 5.5.

The example indicates that the cost-to-serve value of a customer is not merely dependent on its distance from the depot. Rather, it is influenced by various factors, including the locations

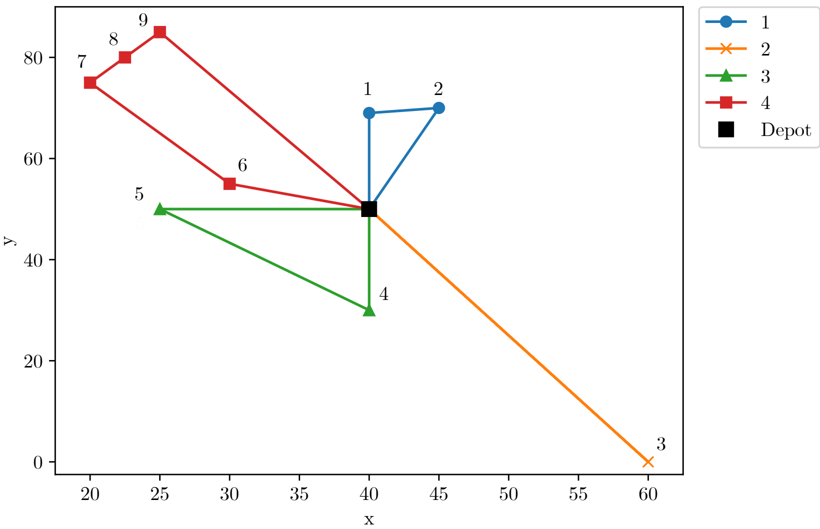


FIGURE 5.13: *Cost-to-serve demonstration — Vehicle route for serving all customers.*

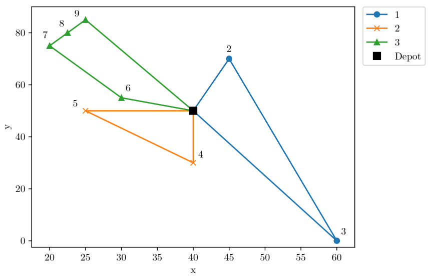
TABLE 5.5: *The routing cost and cost-to-serve for customers in the verification dataset.*

Customer (i)	R_i	c_i
1	\$28.89	\$0.63
2	\$28.75	\$0.77
3	\$18.75	\$10.77
4	\$25.91	\$3.61
5	\$29.18	\$0.34
6	\$33.33	−\$3.81
7	\$29.34	\$0.18
8	\$29.52	\$0.00
9	\$28.63	\$0.89

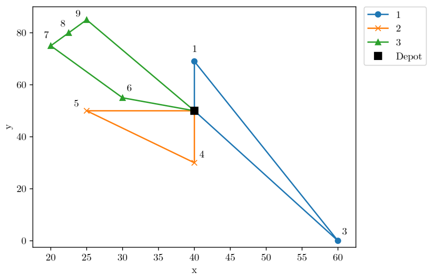
and demands of accompanying customers, the vehicle routing algorithm employed, as well as the capacity of the delivery vehicle.

Two interesting results include that of customer agents 6 and 8. Customer agent 6 has a negative resultant cost-to-serve value, which may seem counter-intuitive. When considering, however, the influence of a specific routing algorithm, the delivery vehicle capacity, and the customer demands, this result becomes plausible. In the original route, illustrated in Figure 5.13, delivery vehicle agent 3 has a cumulative load of 60 as a result of serving customer agents 4 and 5. An additional demand of 20 by customer agent 6 would exceed the vehicle capacity. If customer agent 6 is removed, however, as in the instance shown in Figure 5.14(e), the next customer agent is considered for delivery by delivery vehicle agent 3. Customer agent 7, with a demand of 10, may therefore be added to delivery vehicle agent 3. This results in a route that is significantly less efficient. As such, by removing this specific customer agent, the route becomes more expensive.

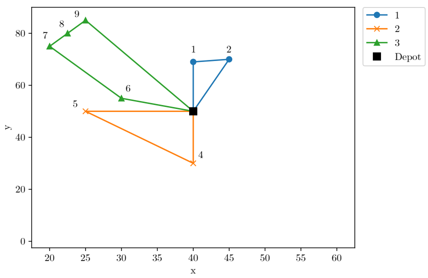
Another interesting case is observed when calculating the cost-to-serve value of customer agent 8. A cost-to-serve value of \$0.00 is calculated, indicating that no additional cost is incurred to serve customer agent 8. This is verified in Figure 5.14(h), as it is clear that the route of delivery vehicle agent 4 does not change as a consequence of removing customer agent 8. The workings



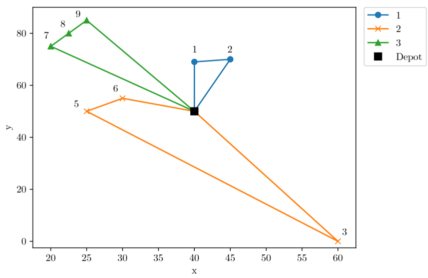
(a) *Route without customer 1*



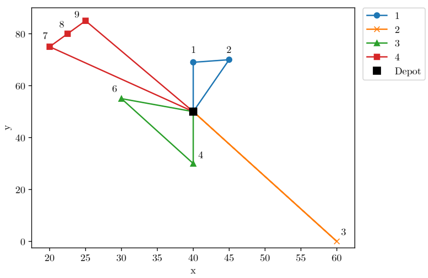
(b) *Route without customer 2*



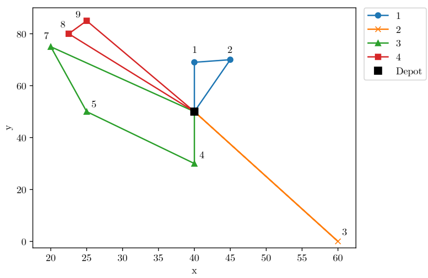
(c) *Route without customer 3*



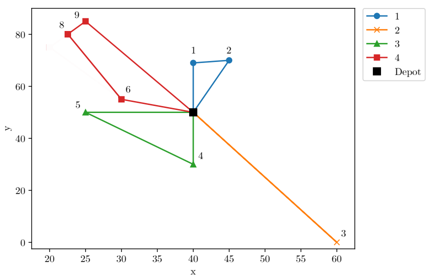
(d) *Route without customer 4*



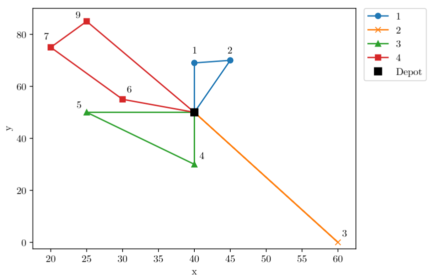
(e) *Route without customer 5*



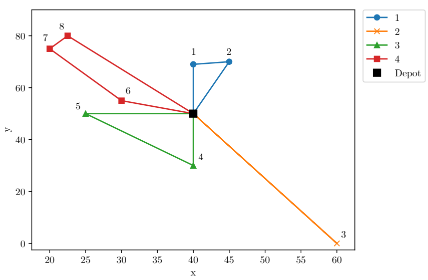
(f) *Route without customer 6*



(g) *Route without customer 7*



(h) *Route without customer 8*



(i) *Route without customer 9*

FIGURE 5.14: *A demonstration of the cost-to-serve algorithm while omitting one customer agent at a time.*

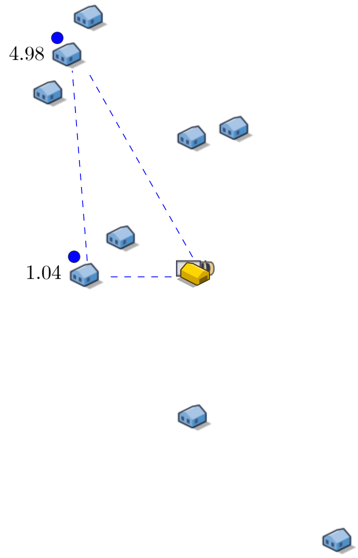
of Algorithm 4.3 is therefore verified through this experiment. Although only demonstrated for the sweep algorithm, Algorithm 4.3 generalises and may be applied to the savings algorithm.

5.4.2 Verification of cost-to-serve recalculation and sorting of customer agents

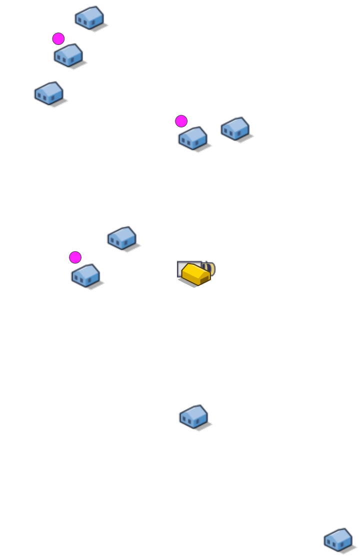
Each time an additional customer agent places an order, the cost-to-serve value of the customer agents already in the `active` state may be influenced. In Experiment 7, it is verified that the cost-to-serve values of arriving customer agents, as well as that of customers agents already in the `active` state, are calculated correctly each time a new order is placed. After verifying that the correct cost-to-serve values are calculated, it is verified that customer agents transition into the correct state, based on the threshold described in §4.5.3.

In verifying these aspects, the verification customer dataset presented in Table 5.4 is implemented in a simulation run. The customer agents place orders over time and their cost-to-serve values and states are observed. The `fixed_incentive` and `loss_aversion` parameters are set equal to $0.50 and 1, respectively. As such, if `cost_to_serve` > 0.5 for a given customer agent, it is expected that the customer agent should transition into the `CCS_candidate` state, otherwise into the `regular_delivery_candidate` state. Similar to Experiment 6, the sweep algorithm is used for demonstration purposes, although the principles remain the same for any routing heuristic applied. Finally, the experiment is conducted with a fixed seed for the random number generator, enabling reproducible simulated scenarios.

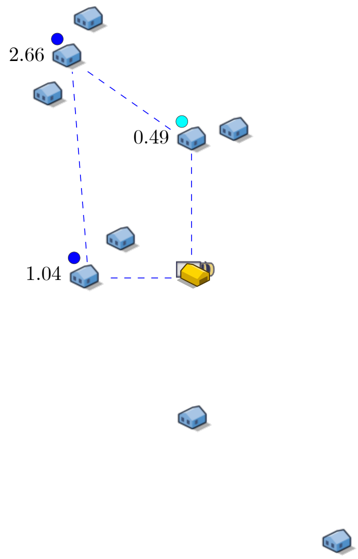
The scenario produced in Experiment 6 is demonstrated in Figure 5.15 with screenshots of the simulation screen over the course of two additional orders being placed. These screenshots show the customer agents in Table 5.4, with the associated `cost_to_serve` variable value of each agent annotated next to its icon (refer to Figure 5.13 for the index of each customer agent). Consider the case where two orders have already been placed, namely those of customer agent 5 and 8. Their cost-to-serve values and the associated delivery vehicle route is shown in Figure 5.15(a). As soon as an additional order is placed, all customer agents transition to the `sort_substate` state, where their `cost_to_serve` variable values are updated prior to their re-sorting into the correct subsequent state. This instance is illustrated in Figure 5.15(b), where customer agent 1 places an order and all customer agents in the `active` state transition to the `sort_substate` state. The updated route and cost-to-serve values of the active customer agents are illustrated in Figure 5.15(c). The newly arrived customer agent 1 has a cost-to-serve value of $0.49, and is therefore sorted into the `regular_delivery_candidate` state, according to the aforementioned threshold. The cost-to-serve value of customer agent 5 remains unchanged after the additional order is placed. For customer agent 8, on the other hand, the additional order results in a decrease in their cost-to-serve value, from $4.98 to $2.66, as the route becomes more effective. It is therefore verified that the cost-to-serve values of customer agents are recalculated and reassigned correctly following the placement of a new order. Furthermore, it is verified that the customer agents are sorted into the correct state, based on their cost-to-serve values and the set threshold. For the final demonstration, customer agent 4 places an order, resulting in an additional delivery vehicle agent required to fulfil all orders. The delivery route, as determined by the sweep algorithm, along with the updated `cost_to_serve` variable values of all customer agents are shown in Figure 5.15(d). The first delivery vehicle agent serves customer agents 1 and 4, with a cumulative demand of 70. The second delivery vehicle agent serves the remainder of the customer agents, with a cumulative demand of 20. It can be seen that customer agent 1 is sorted into the `CCS_candidate` state following an update of its `cost_to_serve` variable value from $0.49 to $4.80. This verifies that at any given instance of the simulation, all customer agents will be in the appropriate state for customer crowd-shipping, based on their current cost-to-serve value and the threshold determined by the model user.



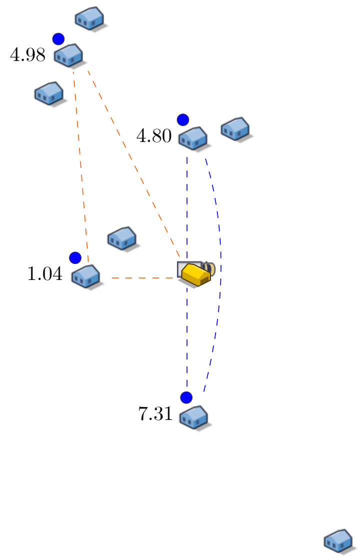
(a) *Cost-to-serve values of two active customer agents*



(b) *Recalculation of cost-to-serve after a new order*



(c) *Cost-to-serve values after a new order*



(d) *Cost-to-serve values after another new order*

FIGURE 5.15: *Verification of the cost-to-serve values with new orders arriving in the system.*

5.5 Verification of OD agent behaviour

To verify the behaviour of OD agents, as described in §4.4 and §4.5, a number of verification tests are performed. First, it is verified that the value of time of the OD agents are assigned as expected. Furthermore, focus is placed on the OD agent decision-making process, relating to the selection of orders to deliver. It is verified that OD agents consider both their value of time and the incentive offered in making the decision that will maximise their perceived gain.

5.5.1 Value of time distribution

The value of time of an individual OD agent is based on their income bracket and the presence of congestion while a trip to the retailer is taken. As described in §4.4.2.2, each OD agent is assigned a `income_level` parameter value, according to the `income_bracket_probabilities` custom distribution, at the initialisation of the model. Thereafter, the `base_value_of_time` parameter value for each OD agent is drawn from a distribution based on their income level. Finally, each time an order is placed, the effect of congestion is taken into account by considering the value of the `peak_hours` schedule and updating the `value_of_time` variable.

In Experiment 8, it is verified that the OD agents are divided into three distinct income levels, governed by the `income_bracket_probabilities` custom distribution. As a continuation, it is verified that the `base_value_of_time` parameter value of the respective income groups are calculated correctly. For Experiment 8, the probabilities captured in the `income_bracket_probabilities` custom distribution is set to a uniform discrete distribution, such that $p_{low} = p_{mid} = p_{high} = \frac{1}{3}$. Furthermore, the `minimum_VOT` parameter value is set to \$4.00 per hour, while the variance of all populations are set to 0 (*i.e.* $\sigma_{low} = \sigma_{mid} = \sigma_{high} = 0$). A simulation run is initialised with a fixed seed for the random number generator in order to replicate the run in the subsequent experiments. Furthermore, in order to demonstrate the value of time adequately, the population size of OD agents are set to 100. The resultant distribution of the `base_value_of_time` parameter value for the `OD` agent population is shown in Figure 5.16.

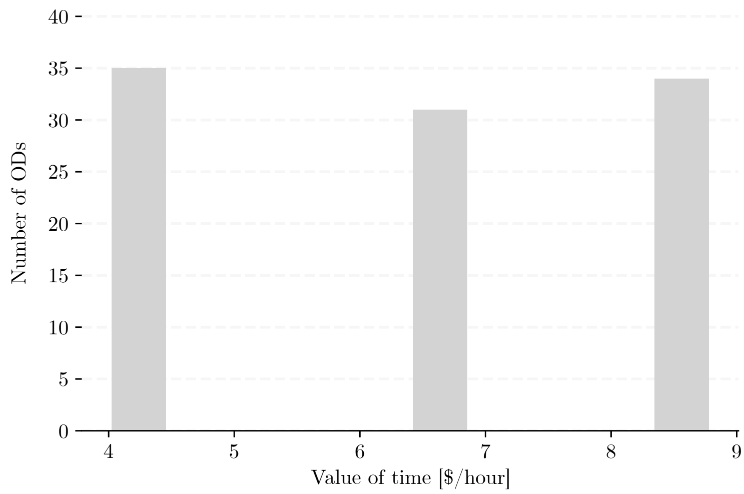


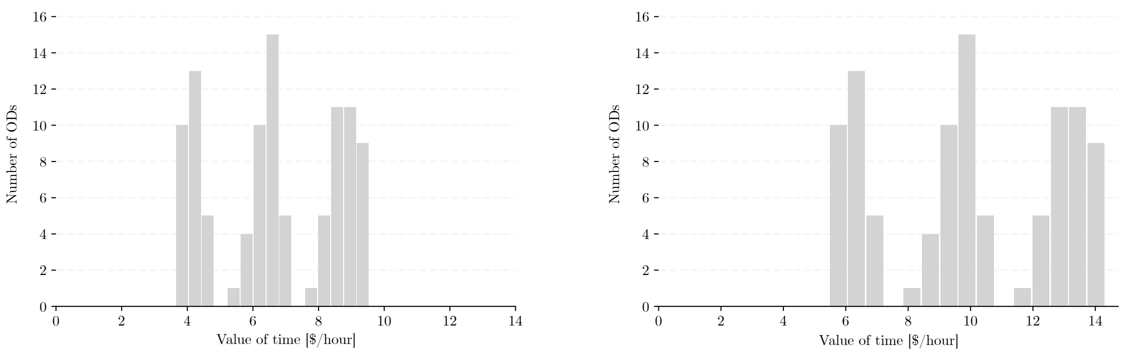
FIGURE 5.16: *The distribution of base value of time parameter values for 100 OD agents.*

The first observation is that the value of time of the population is separated into three roughly equal parts. In particular, there are 34 OD agents in the low income bracket, 31 OD agents in the middle income bracket, and 35 OD agents in the high income bracket. This verifies that the division into income brackets follows the probabilities of the `income_bracket_probabilities` custom distribution. Furthermore, it can be seen that the `base_value_of_time` parameter value is calculated correctly for the three classes where the variance of all classes are set to 0 (*i.e.*

all values take the value μ). In particular, all OD agents in the low income bracket takes a value equal to the `minimum_VOT` parameter value, which is set to \$4.00 per hour. The OD agents in the middle income bracket, on the other hand, is set equal to 1.6 times the `minimum_VOT` parameter value, equating to \$6.40 per hour. Finally, all OD agents in the middle income bracket is set equal to 2.2 times the `minimum_VOT` parameter value, equating to \$8.80 per hour.

In Experiment 9, it is verified that when calculating the `base_value_of_time` parameter value, the element of stochasticity introduced by the population variance is reflected adequately. Furthermore, it is verified that the effect of congestion is implemented correctly during the specific congested hours. For demonstration purposes, the demand factor, captured in the `peak_hours` schedule, is set to a value 1.5 during the hours of 16:00–17:00 and to a value of 1 otherwise. In order to compare the results with and without the element of stochasticity, the same fixed seed utilised for the random number generator in Experiment 8 is implemented in Experiment 9. Furthermore, the `minimum_VOT` parameter is again set to a value of \$4.00 per hour and a population of 100 OD agents is evaluated.

When increasing the variance of all populations, three separate distributions are expected, relating to each income bracket. To demonstrate this, in Experiment 9, the variance of all income classes are set equal such that $\sigma_{high} = \sigma_{mid} = \sigma_{low} = 0.4$. The resulting distribution of the `value_of_time` variable values is illustrated in Figure 5.17. Moreover, the distribution is shown for non-peak hours in Figure 5.17(a), whereas the distribution during peak hours is illustrated in 5.17(b).



(a) *Value of time distributiuon without congestion* (b) *Value of time distribution during congestion*

FIGURE 5.17: *Verification of the OD population value of time distribution amongst income brackets.*

When comparing the results of Experiment 8 with that of Experiment 9, the effect of an increased variance is evident. Rather than a discrete value for the `base_value_of_time` parameter, three distributions resembling a normal distribution are formed. The mean value of time for each income bracket, however, remains similar as expected. Furthermore, when comparing the distributions illustrated in Figure 5.17(a) with that of Figure 5.17(b), the effect of the `peak_hours` schedule is evident, as the `value_of_time` variable values increase significantly.

5.5.2 Offer evaluation and acceptance

The modelling of OD decision making, as described in §4.5.4, is considered for verification. Consider the scenario where an OD is about to make a trip to the retailer. The OD agent enters the `shopping` state and is presented with the list of customer agents that are in the

CCS_candidate state. Through the use of a crowdsourcing platform, the OD agent has knowledge of these customer agents' location and therefore knows the required deviation in their route. Furthermore, each OD agent estimates the additional time that such a deviation would require, which is influenced by their modelled average_speed parameter value. Finally, alongside the list of locations, the OD agent is presented with an incentive associated with each customer agent's delivery.

To verify this aspect of OD behaviour, the R1 Solomon [183] problem set for 25 customers, as illustrated in Figure 5.4 is implemented for a simulation run and the behaviour of a particular OD agent is analysed. In particular, Experiment 10 considers the scenario where an OD agent has a V value_of_time variable value of $5.00 per hour and an average_speed parameter value of 10 kilometres per hour. The fixed_incentive and variable_rate parameters are set to values of $1.00 and $0.20 per kilometre, respectively.

In Figure 5.18, a screenshot of the simulation screen during the execution of Experiment 10 is shown at the instance when OD agent j, circled in red, transitions into the shopping state. As the agent commences on their trip to the store, a list of the five customer agents in the CCS_candidate state is presented to the OD agent.

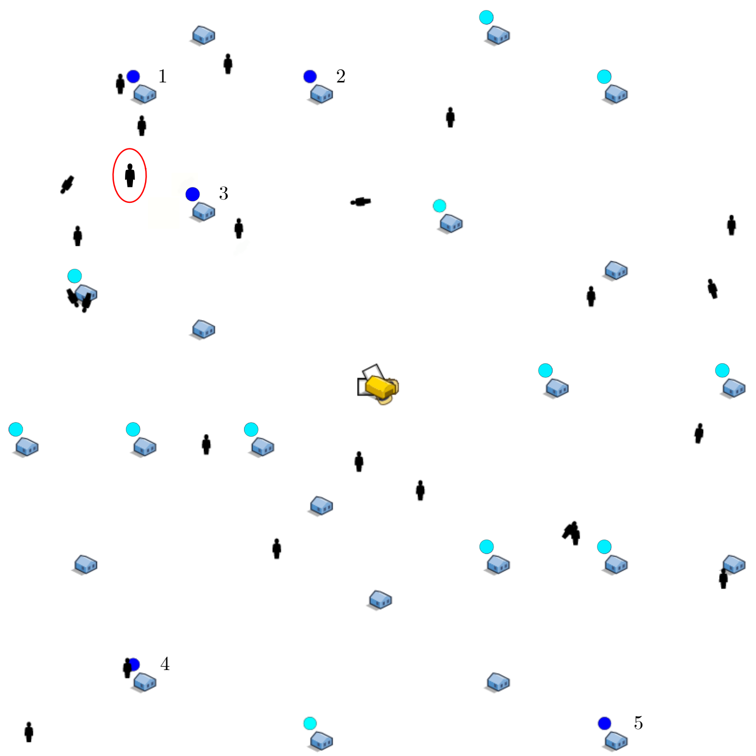


FIGURE 5.18: *Screenshot of the simulation screen of Experiment 10 before the OD agent selects an offer.*

The evaluation of the five delivery options for OD agent j is presented in Table 5.6. First, the deviation δ_{ij} required for OD agent j to deliver to customer agent i, as well as the associated incentive offered λ_{ij}, is shown. Furthermore, the estimated time required to deliver to each customer agent t_{ij}, as well as the associated value of the deviation time v_{ij}, is shown. Finally, the perceived gain g_{ij} is calculated as the difference between the incentive and the value of time for each customer agent. These values are captured for the purpose of verification for the instance shown in the Figure 5.18.

TABLE 5.6: *The delivery candidates for OD agent j.*

i	δ_{ij} [km]	λ_{ij} [\$]	t_{ij} [min]	v_{ij} [\$]	g_{ij} [\$]
1	0.768	1.154	4.61	0.384	0.770
2	1.436	1.287	8.62	0.718	0.569
3	0.321	1.064	1.93	0.1605	0.904
4	4.576	1.915	27.46	2.288	−0.373
5	7.080	2.416	42.48	3.540	−1.124

The perceived gain of delivering to customer agents 4 and 5 are negative, indicating that the OD agent will reject these offers according to the workings of the model. From the remaining offers, however, OD agent j will select the offer that results in the highest perceived gain, namely that of delivering to customer agent 3. OD agent j will therefore travel an additional distance of 0.321 kilometre and receive an incentive of \$1.92. A screenshot of the simulation screen after the OD agent has evaluated the offers is shown in Figure 5.19, verifying the selection of the offer associated to customer agent 3.

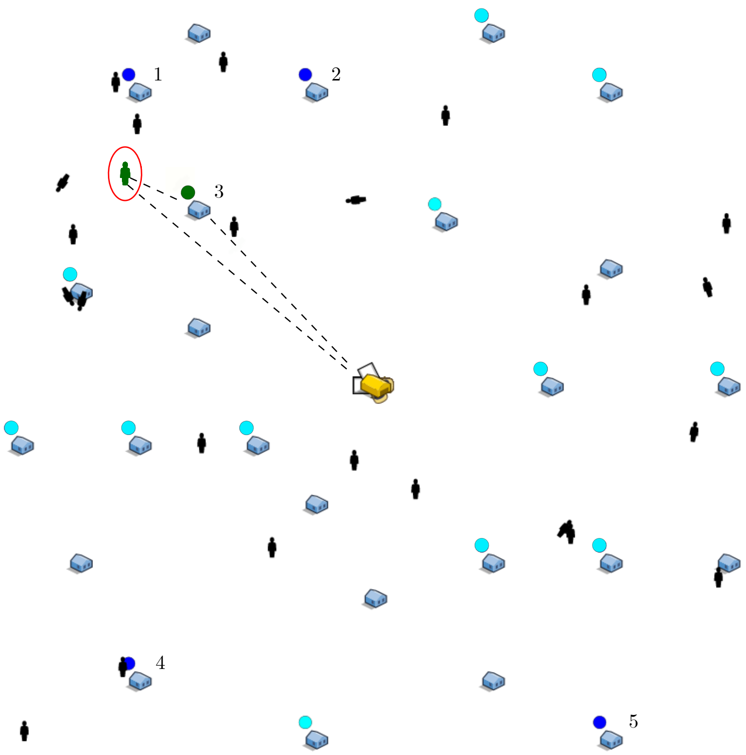


FIGURE 5.19: *Screenshot of the simulation screen of Experiment 10 after the OD agent selects an offer.*

From the results a number of aspects are verified. First, it verifies that OD agent j selects the offer that maximises their personal perceived gain. Furthermore, it is verified that as soon as this offer is accepted, the OD agent transitions to the ■ `performing_delivery` state, while the respective customer agent transitions to the ● `awaiting_CCS_delivery` state.

5.6 Verification of accrued savings

The accrual of company savings through the use of crowd-shipping is discussed in §4.6.2. The monetary value of savings is recorded at each set of company deliveries, whereas the percentage delivery savings is calculated at the termination of a simulation run. In order to verify the correct calculation of the monetary delivery savings, the state of the system is considered in the interval between two sets of company deliveries.

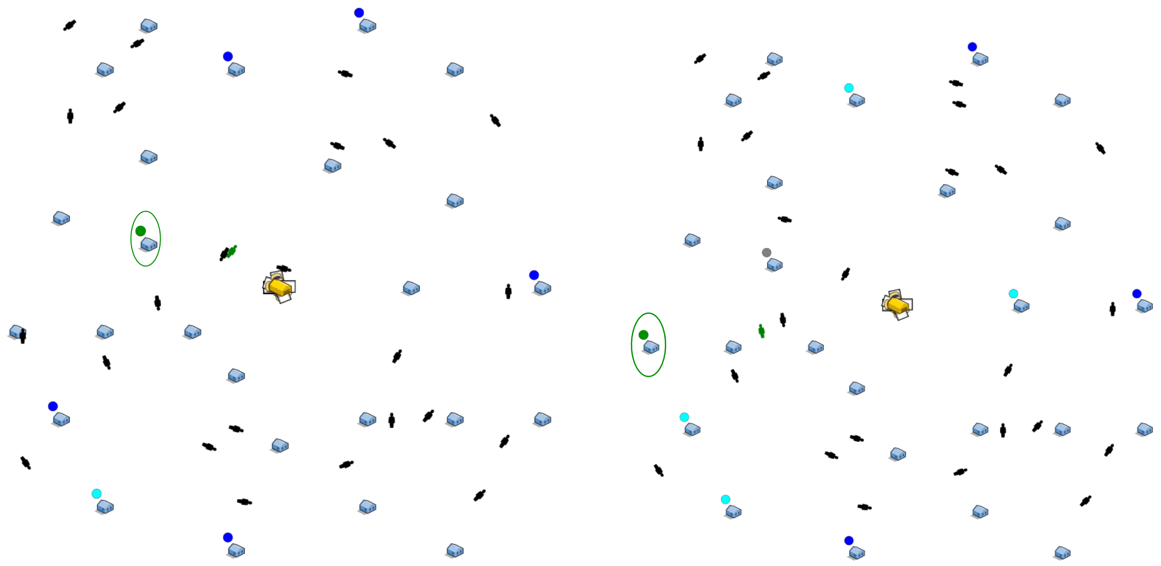
In verifying the correct calculation of the delivery savings, Experiment 11 considers a simulation run with a fixed seed for the random number generator and the R1 Solomon [183] problem set for 25 customers serving as input. The time period between the company deliveries at 11:00 and at 17:00 is considered for a `deliveries_per_day` parameter value of 2 deliveries per day. The extreme case is considered where the `base_order_rate` parameter value is set to 6 orders per day, ensuring all customer agents will place an order during this time. Furthermore, the incentive scheme and loss aversion is considered with the `fixed_incentive` parameter value set to \$1.00, the `variable_rate` parameter value set to \$0.20 per kilometre, the `max_incentive` parameter value set to \$3.00, and the `loss_aversion` set to 1.

During this interval two successful instances of crowd-shipping occurred, as illustrated in the screenshots in Figure 5.20. For the first instance of OD delivery, as illustrated in Figure 5.20(a), an OD agent with a V `value_of_time` variable value of \$4.25 per hour commenced on their journey, after being offered an incentive of \$4.93 to deliver to the customer agent circled in green. The value of the deviation time was perceived as \$2.05, resulting in a perceived gain of \$2.88 for the OD agent. At the moment in time at which the trip was outsourced, the customer agent had a cost-to-serve value of \$5.88, which is \$0.95 more expensive than the incentive paid. As such, it is predicted that the company would save \$0.95 by outsourcing the delivery of this order.

The second OD delivery is illustrated in Figure 5.20(b), with the outsourced customer agent circled in green. The incentive offered was \$1.70, while the OD agent's perceived value of the deviation time was \$0.52, resulting in a perceived gain of \$1.18 for the OD agent. In this instance, however, the cost-to-serve value of the outsourced customer agent was \$1.50, resulting in predicted loss of \$0.20 to the company.

The results of Experiment 11 exemplifies that there is no guarantee on the savings of an individual OD delivery. Rather, the aim is to merely reduce the number of instances where a loss is incurred by classifying customer agents based on their cost-to-serve values as described in §4.5.3. The full savings or losses cannot be determined in the moment, but can only be fully known after-the-fact, once the next company delivery is performed. As described in §4.6.2, the company savings is determined by first calculating the baseline delivery cost (*i.e.* the delivery cost if no crowd-shipping occurred from the previous company delivery). Thereafter the actual delivery cost is calculated by adding the incentives associated to the customer agents in the `CCS_complete` state to the cost of serving the customer agents in the `awaiting_regular_delivery` state.

In Figure 5.21, a screenshot of the simulation screen during Experiment 11 is shown at the instance before delivery at 17:00. The customer agents that have been served by OD agents have transitioned to the `CCS_complete` state as depicted by the customer agents circled in grey. The baseline delivery route is the same as that depicted in Figure 5.11 and results in a cost of \$415.77. The actual delivery route, as depicted in Figure 5.21, has a cost of \$414.68, while the total incentives paid amount to \$6.63. The actual total delivery cost therefore amounts \$421.31, with the company delivery saving amounting to a loss of \$5.54.



(a) *First instance of crowd-shipping* (b) *Second instance of crowd-shipping*

FIGURE 5.20: *Screenshots of the simulation screen during Experiment 11 illustrating two instances of customer agents served by OD agents.*

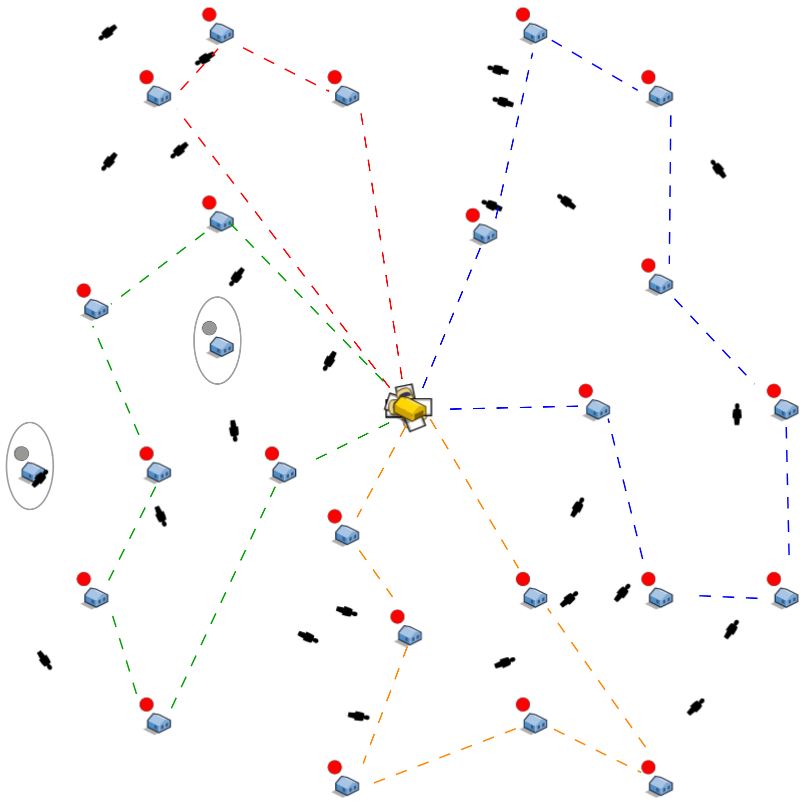


FIGURE 5.21: *A screenshot of the simulation screen during Experiment 11 illustrating the delivery route after two customer agents have been served by OD agents.*

The calculations of the company delivery savings are verified as correct. It is also important to note and summarise the high level of complexity involved in this calculation. In Experiment 11 it is demonstrated that the savings is dependent on a number of factors. First, the combination of values of the `loss_aversion` and `fixed_incentive` parameters determine which customer agents are candidates for crowd-shipping, aiming to only consider those with high `cost_to_serve` values. The `cost_to_serve` values, in turn, are dependent on the number of customers agents in the `active` state, their associated demand, and their geographical distribution, as well as the `cost_per_distance` parameter value. Furthermore, the combination of the `fixed_incentive`, `variable_rate` and `max_incentive` parameters, in addition to the OD agent's `value_of_time` variable value, determines the magnitude of incentives offered and the willingness to accept the offers. Finally, the number and distribution of customer agents that place orders during the duration of the inter-delivery period influences the resultant savings.

5.7 Chapter summary

In §5.1, the need for verification was communicated and a number of verification techniques were outlined. The first verification tests involved the customer order frequency in §5.2. Thereafter, the verification of vehicle routing and last-mile deliveries were documented in §5.3. Furthermore, the verification of the cost-to-serve algorithm was documented in §5.4, both with respect to the details of the algorithm, as well as its implementation in the simulation model. The behaviour of OD agents were considered in §5.5, verifying the value of time implementation. Finally, the delivery savings calculation was verified in §5.6.

CHAPTER 6

Model evaluation

Contents

In this chapter the proposed model as developed in Chapter 4 and verified in Chapter 5, is further validated and evaluated. In particular, the output of the model (*i.e.* the KPIs discussed in §4.6) is evaluated under a number of assumptions to generate deeper insight into the model and customer crowd-shipping in general. The aim of the chapter is to foster confidence in the model, establish reasonable values for the various input parameters, and evaluate the performance of customer crowd-shipping under various conditions. This is achieved through the use of parameter variation, sensitivity analysis, as well as an evaluation of the model under a number of scenarios, and validation from subject matter experts. In §6.1, the parameter variation conducted is discussed, granting insight into the effect of the decisions made by the retailer. In §6.2, on the other hand, the model's sensitivity to a number of assumptions is analysed. Furthermore, a number of scenarios are proposed in §6.3. The model output for these scenarios are analysed and discussed, with particular focus on the saving in delivery cost and reduction in waiting time resulting from customer crowd-shipping. Finally, in §6.4, the validation performed through subject matter experts is discussed.

6.1 Parameter variation

Parameter variation is conducted to determine how user-defined parameters influence the output of the simulation model. In particular, the parameters that should be defined by the retailer in question, as discussed in §4.7.1.1, are considered individually. This includes the `deliveries_per_day` parameter, the set of parameters comprising the incentive scheme, as well as the `loss_aversion` parameter. Experiments are conducted over a simulated time-frame of two weeks under a particular set of conditions, while varying one of the parameters per experiment and observing the effect on the relevant KPIs. Furthermore, multiple replications of the experiments are conducted and a variable random seed is used, resulting in multiple independent runs. This approach is taken in an attempt to isolate, as best possible, the effect of a particular parameter and minimise the influence of other random variables in the simulation. The KPIs considered for a particular experiment may include one or more metrics of the output datasets, as discussed in §4.6. Moreover, the results of the parameter variation may serve as a partial validation of the simulation model, as discussed in §3.2.2. By observing the influence of the aforementioned user-defined parameters and confirming it to be expected and sensible, additional confidence may be fostered in the proposed model.

As mentioned in §4.2.1, the model utilises theoretical data pertaining to the location and demands of customers, the location of the depot, and the vehicle capacities. Similar to Archetti *et al.* [9], the evaluation of the model considers the benchmark instances of the CVRP as proposed by Solomon [183]. In particular, the parameter variation experiments are performed utilising problem set R1 for 100 customers randomly distributed around the depot. The geographical distribution of customers around the depot is illustrated in Figure 6.1, with additional information on the problem sets provided in Appendix A.

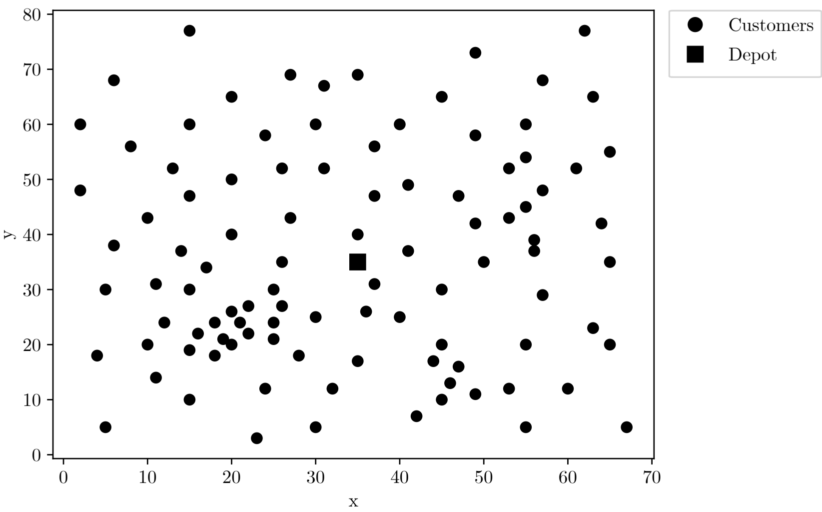


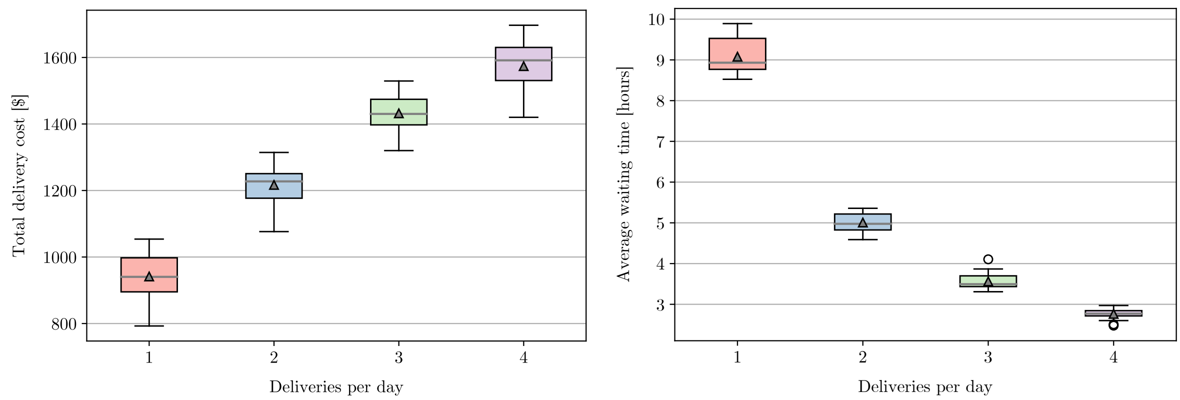
FIGURE 6.1: *The locations of customers and the depot according to the R1 problem set from Solomon [183].*

6.1.1 Deliveries per day

The first set of experiments conducted involves varying the number of deliveries per day and analysing the effect on the relevant KPIs. In particular, the average waiting time and the total delivery cost are considered while varying the `deliveries_per_day` parameter value. In analysing the effect of the delivery frequency in isolation and establishing a baseline cost and waiting time, the use of ODs are not considered for this experiment. Furthermore, as the baseline performance of these KPIs are dependent on the CVRP solution methodology, the experiment is replicated for the sweep and savings algorithms, respectively.

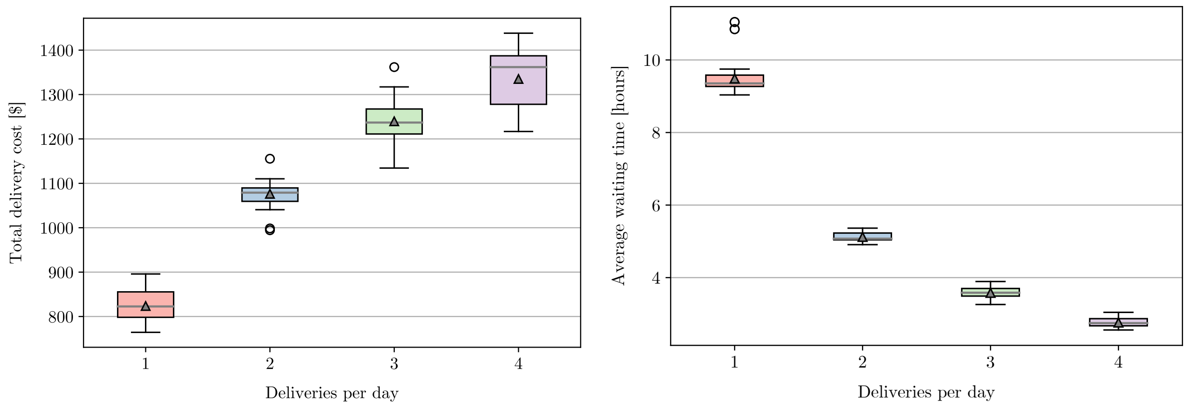
For Experiment 12, the `deliveries_per_day` parameter is varied from a value of 1–4 deliveries per day in increments of 1. Furthermore, the `cost_per_distance` and `base_order_rate` parameters are set to the respective values of \$2.00 per kilometre and 0.25 orders per day. Finally, the sweep algorithm is utilised Experiment 12.1, whereas the savings algorithm is utilised in Experiment 12.2.

For both experiments, the simulation is run for a simulated time of two weeks, to observe the effect on the relevant KPIs. Specifically, the sum of all entries in the `delivery_vehicle_costs` dataset, as well as the mean of all the entries in the `waiting_time` dataset, for a given replication are considered. Each simulation configuration is replicated 30 times with a variable random seed resulting in independent runs. Box-plots of the relevant KPIs for Experiment 12.1 (utilising the sweep algorithm) are shown in Figure 6.2, where each point represents the output of a single replication. The total delivery cost per replication is shown in Figure 6.2(a), whereas the average waiting time per replication is shown in Figure 6.2(b). Furthermore, the results of Experiment 12.2 (utilising the savings algorithm) are shown in Figure 6.3. In particular, box-plots of the total delivery vehicle cost per replication is shown in Figure 6.3(a), whereas box-plots of the average waiting time per replication are shown in Figure 6.3(b). In comparing the output of Experiment 12.1 and Experiment 12.2, the mean results across all replications for each configuration are shown in Table 6.1.



(a) *Boxplot of the total delivery vehicle cost* (b) *Boxplot of the average waiting time*

FIGURE 6.2: *The resultant KPIs for 30 simulation replications while varying the deliveries per day utilising the sweep algorithm.*



(a) *Boxplot of the total delivery vehicle cost* (b) *Boxplot of the average waiting time*

FIGURE 6.3: *The resultant KPIs for 30 simulation replications while varying the deliveries per day utilising the savings algorithm.*

From Figure 6.2 and Figure 6.3, it is evident that the delivery frequency has a significant influence on both the delivery cost and the average waiting time. In particular, from the results illustrated in Figure 6.2(a) and Figure 6.3(a), it can be seen that a more frequent delivery schedule results in a higher delivery vehicle cost. Conversely, from the results illustrated in Figure 6.2(b) and

TABLE 6.1: *The mean total delivery cost and mean average waiting time for 30 replications varying the deliveries per day for the savings and sweep heuristic.*

Deliveries per day	Savings heuristic Delivery cost [$]	Savings heuristic Waiting time [hours]	Sweep heuristic Delivery cost [$]	Sweep heuristic Waiting time [hours]
1	824	9.49	942	9.08
2	1076	5.12	1217	5.01
3	1240	3.58	1432	3.56
4	1335	2.76	1574	2.77

Figure 6.3(b), it is evident that the average waiting time decreases as deliveries are performed more frequently. It is further noted that the average waiting time decreases significantly when the delivery frequency is adjusted from one to two deliveries per day. This change in average waiting time seems to diminish with the subsequent increments of delivery frequencies. Finally, it is noted that the objectives are conflicting — a reduction in average waiting time is associated with an increased delivery cost when utilising the regular company delivery vehicles.

The effect of varying the frequency of delivery is found to have similar effects when utilising the savings or sweep algorithm as a solution to the CVRP. It is furthermore evident from the results in Table 6.1 that the savings algorithm outperforms the sweep algorithm both with respect to delivery cost and average waiting time. The remainder of the parameter variation experiments, however, are conducted while using the sweep algorithm. It is decided that the worse-performing algorithm is investigated initially in a conservative approach, aiming to reflect the real-world system, in order to gain a better understanding of the customer crowd-shipping system.

To effectively evaluate the effects of customer crowd-shipping on a retailer's last-mile delivery system, the configuration of the regular deliveries needs to remain constant. Thus, it is necessary to select a default value for the `deliveries_per_day` parameter for the remainder of the model evaluation. To this end, the results of Experiment 12 are considered both with respect to the increasing consumer expectations of short lead-times for last-mile deliveries, as mentioned in §2.2.1, as well as the need to demonstrate the effects of customer crowd-shipping adequately. An average waiting time of greater than eight hours, as in the case where the `deliveries_per_day` parameter is set to a value 1 delivery per day, may be considered unreasonably high. On the other hand, the effects of crowd-shipping may be most prominently observed if there are adequate opportunities for ODs to serve waiting customers. In considering these aspects, a default `deliveries_per_day` parameter value of 2 deliveries per day is proposed for the remainder of the model evaluation. This provides a reasonable baseline waiting time and delivery cost, while granting sufficient opportunity to observe the effects of customer crowd-shipping. Finally, in maintaining the expected order frequency from customer agents, it is suggested, as mentioned in §5.2.1, to adopt a `base_order_rate` parameter value that is less than half of the `deliveries_per_day` parameter value. As such, a default `base_order_rate` parameter value of 0.25 orders per day is employed for the remainder of the model evaluation.

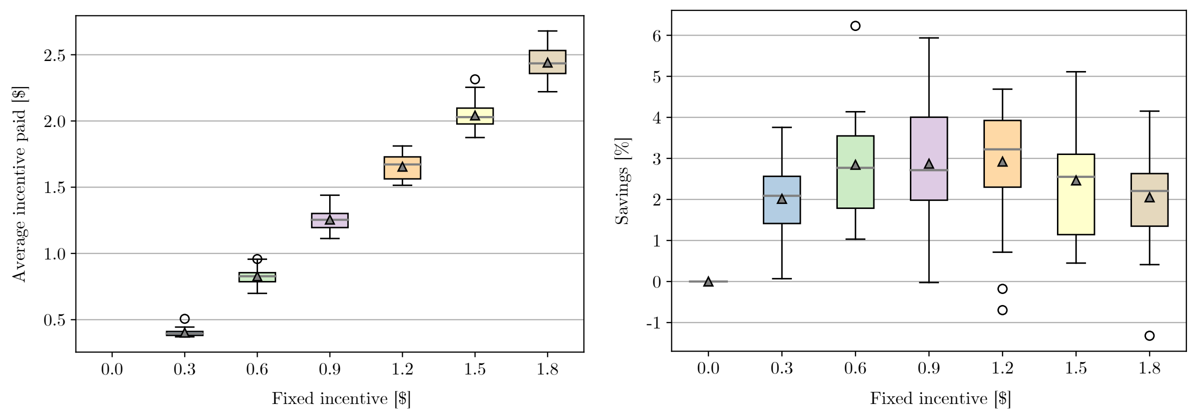
6.1.2 Validation of the incentive scheme values

The next set of experiments involves varying the values that comprise the incentive scheme, as described in §4.5.2. In particular, the `fixed_incentive` and `variable_rate` parameter values are varied respectively, to find the range within which these parameters are deemed valid.

Additionally, in considering the experiment results, the need for a maximum incentive parameter is motivated. Finally, a sensible default value of the `max_incentive` parameter is derived and motivated based on the experiment results.

The KPIs considered in these experiments include the average cost of incentives paid, as well as the percentage savings accrued. The sweep algorithm is utilised as the CVRP solution methodology throughout the execution of the various simulation runs. The `OD_customer_ratio` parameter is set to a value of 1 OD per customer, whereas the `loss_aversion` parameter is set to a value of 1. Furthermore, the `deliveries_per_day` and `cost_per_distance` parameters are set to their respective default values of 2 deliveries per day and $2.00 per kilometre. Additionally, the `minimum_VOT` parameter value is set to $5.00 per hour. Finally, the order rate parameters, namely `base_order_rate` and `base_OD_rate`, are set to the values of 0.25 and 1 order per day, respectively.

In Experiment 13.1, the `fixed_incentive` parameter is varied from a value of $0.00 to $1.80 in increments of 0.3. The `variable_rate` parameter, on the other hand, is kept constant at $0.30 per kilometre. In order to observe the workings of the incentive scheme before the introduction of a maximum incentive, the `max_incentive` parameter is set to a large value of 50. The simulation is run for a simulated time of two weeks, to observe the effect on the relevant KPIs. Specifically, the mean of all the entries in the `incentives_paid` dataset, as well as the `percentage_savings` variable value are considered for each replication. The simulation is replicated 30 times with a variable random seed resulting in independent runs. A box-plot of the relevant KPIs are shown in Figure 6.4, where each point represents the output of a single replication. The average value of incentives paid per replication is shown in Figure 6.4(a), whereas the delivery savings per replication is shown in Figure 6.4(b).

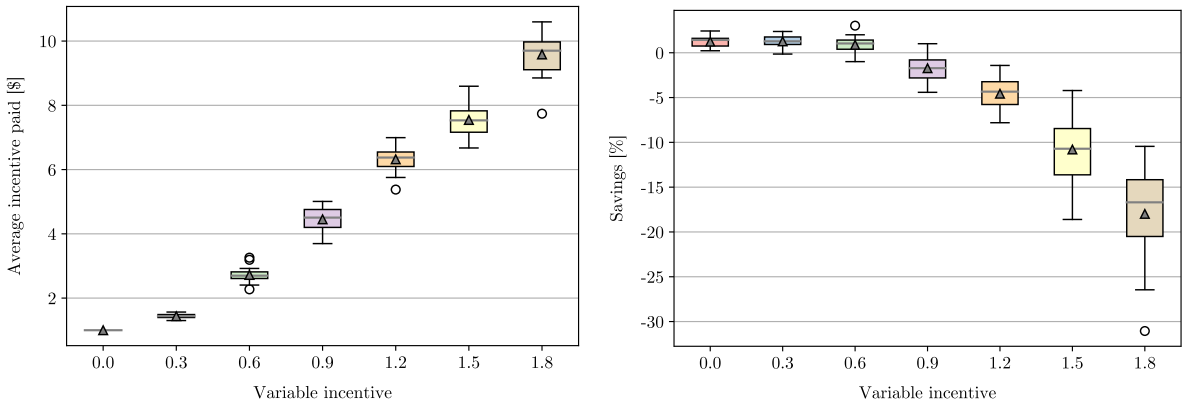


(a) *Boxplot of the average incentive paid* (b) *Boxplot of the percentage saving*

FIGURE 6.4: *The average incentives and percentage savings for 30 simulation replications while varying the fixed incentive.*

The results illustrated in Figure 6.4(a) indicate a near-linear increase in the average incentives paid as the fixed incentive is increased. This is expected, given the manner in which the offered incentive is calculated. The resultant savings distribution, illustrated in Figure 6.4(b), provides more interesting results. The initial increase in the fixed incentive from $0.00 to $0.60, results in an increase in savings. This initial increase in savings is explained by the larger offered incentive resulting in more accepted offers (*i.e.* more OD agents perceive the offers as beneficial). The savings resulting from the increasing number of accepted trips is countered, however, by the increasing cost of the incentives paid. For a value of the fixed incentive greater than $1.20, there is an associated decrease in savings. As such, it is expected that there is a balance to be struck with respect to making the offers sufficiently large to ODs, without the total incentives paid nullifying the decrease in delivery cost. Considering these results, it is recommended that the proposed `fixed_incentive` parameter value should fall in the range of \$0.60 to \$1.20 for this model.

In Experiment 13.2, the `variable_rate` parameter is varied from a value of \$0.00 to a value of \$1.20 per kilometre in increments of 0.3. The `fixed_incentive` parameter is set to a fixed value of \$1.00. Similar to the previous experiment, each replication is run for a simulated time of two weeks, while considering the mean of all the entries in the `incentives_paid` dataset, as well as the `percentage_savings` variable value. The simulation is replicated 30 times with a variable random seed resulting in independent runs. A box-plot of the relevant KPIs are shown in Figure 6.5, with the average value of incentives paid per replication illustrated in Figure 6.5(a) and the delivery savings per replication illustrated in Figure 6.5(b).



(a) *Boxplot of the average incentive paid* (b) *Boxplot of the percentage saving*

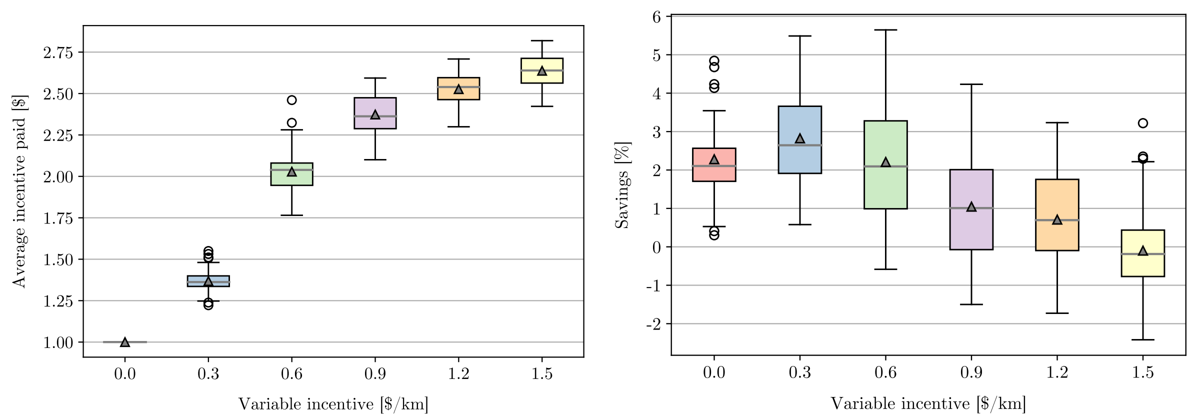
FIGURE 6.5: *The average incentives and percentage savings for 30 simulation replications while varying the variable rate.*

From the results shown in Figure 6.5(a), it is clear that the average incentive paid increases with an increasing `variable_rate` parameter value. The rate of increase is greater than that resulting from the increase in the `fixed_incentive` parameter value. This implies that the cost of incentives paid is more sensitive to changes in the `variable_rate` parameter value than to changes in the `fixed_incentive` parameter value. Furthermore, from Figure 6.5(b), it is clear that the savings are highly sensitive to the variable rate of the incentive scheme. For `variable_rate` parameter values greater than \$0.60, substantial losses are observed. This is as a result of offered incentives becoming exorbitantly high. With such high incentives, ODs are increasingly likely to perform deliveries that involve major deviations in their route. In addition to the negative effect on company savings, such large deviations may be seen as contrary to the idea of ODs delivering "on their way home" thereby abandoning the focus of collaboration in local communities.

A number of potential solutions exist to solve the problem of exorbitantly high incentives and unrealistically long trips. The first potential solution involves limiting the candidates presented to an OD to those that reside within a given deviation for the OD in question. This approach, however, may oppose the crowdsourcing principle of an open-call to work by unnecessarily limiting the options provided to ODs. Furthermore, to filter the candidates each time an OD announces a trip may introduce significant complexity to the model. An alternative approach involves the introduction of a maximum incentive parameter, which limits the value of incentives offered to ODs. This parameter refers to the `max_incentive` parameter, introduced in §4.5.2. By following this approach, exorbitantly high incentives and the resulting company losses may be avoided. Moreover, each OD is still presented with the full set of customers that are being outsourced by the retailer.

In order to find a default value for the `max_incentive` parameter, the results illustrated in Figure 6.5 are considered. Given that the purpose of this parameter is to limit losses by restricting the value of incentives, it is necessary to find the point at which losses are incurred. From Figure 6.5(b), it is clear that significant losses occur for cases where the `variable_rate` parameter value is greater than \$0.60 per kilometre. Moreover, from Figure 6.5(a), it can then be deduced that these losses result from the cases where the average incentive is greater than \$3.00. As such, a default value of \$3.00 is proposed for the `max_incentive` parameter in this model.

To perceive the effect of introducing a maximum incentive value, the experimental configuration of Experiment 13.2 is replicated. In Experiment 13.3, however, the `max_incentive` parameter is set to a value of \$3.00. The simulation is replicated 30 times with a variable random seed resulting in independent runs. A box-plot of the relevant KPIs are shown in Figure 6.6, where each point represents the output of a single replication. The average value of incentives paid per replication is shown in Figure 6.6(a), whereas the percentage delivery savings per replication is shown in Figure 6.6(b).



(a) *Boxplot of the average incentive paid*   (b) *Boxplot of the percentage saving*

FIGURE 6.6: *The average incentives and percentage savings for 20 simulation replications while varying the variable rate after the introduction of a maximum incentive parameter.*

The first observation made is that the average incentive paid is limited by the `max_incentive` parameter, as illustrated in Figure 6.6(a). Rather than the near-linear increase seen in Figure 6.5(a) (reflecting the output of Experiment 13.2), the resultant average incentive is restricted as the `variable_rate` parameter value is increased. Furthermore, the resulting effect on the savings can be observed in Figure 6.6(b). For cases where the `variable_rate` parameter value exceeds \$0.60 per kilometre, there seems to be a decrease in the savings accrued. When comparing these results to that obtained in Experiment 13.2, however, the losses incurred are notably less acute. Furthermore, the system's sensitivity to increases in the `variable_rate` parameter value is substantially reduced. The introduction of a maximum incentive parameter may therefore result in a more robust customer crowd-shipping system, with a lower probability of substantial losses incurred as a result of exorbitantly high incentives. Finally, considering the resultant savings shown in Figure 6.6(b), it is proposed that the `variable_rate` parameter should fall in the range from \$0.00 to \$0.60 per kilometre, and take a default value of \$0.30 per kilometre.

6.1.3 Loss aversion parameter

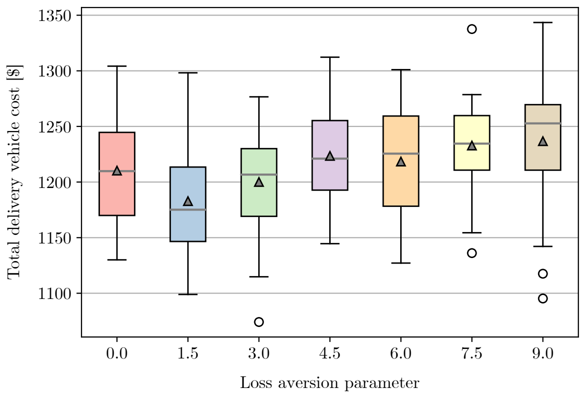
The final experiment of the parameter variation section involves an investigation of the loss aversion parameter, introduced in §4.5.3. The value of this parameter may be considered a company decision, as the retailer has control over which customer agents are classified as candidates for OD delivery. In the proposed model, the `loss_aversion` parameter influences the threshold that governs the classification of customer agents as OD candidates. This experiment serves as an investigation of how the `loss_aversion` parameter ultimately influences the output of the model. In particular, it is investigated how the various KPIs are influenced when changing from a liberal (*i.e.* a low value of the `loss_aversion` parameter) to a more conservative (*i.e.* a high value of the `loss_aversion` parameter) approach to customer crowd-shipping.

For Experiment 14, the `loss_aversion` parameter is varied from a value of 0–9 in increments of 1.5. The availability of ODs are governed by the `OD_customer_ratio` and `base_OD_rate` parameters, which are set to values of 1 OD per customer and 1 order per day, respectively. Additionally, the `minimum_VOT` parameter value is set to \$5.00 per hour. Furthermore, the default incentive scheme is applied, with the `fixed_incentive`, `variable_rate`, and `max_incentive` parameter values set to \$1.00, \$0.30 per kilometre, and \$3.00, respectively. Finally, the `cost_per_distance` and `base_order_rate` parameters are set to their respective default values of \$2.00 per kilometre and 0.25 orders per day.

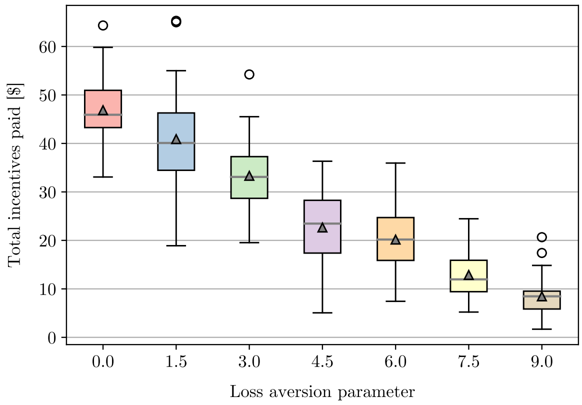
The simulation is run for a simulated time of two weeks, to observe the effect on the relevant KPIs. First, the sum of all delivery costs are considered, with respect to both the `delivery_vehicle_costs` dataset, as well as the `incentives_paid` dataset. Furthermore, the mean of all entries in the `waiting_time` dataset, as well as the value of the `percentage_savings` variable are considered. The simulation is replicated 30 times with a variable random seed resulting in independent runs. A box-plot of the relevant KPIs are shown in Figure 6.8, where each point represents the output of a single, independent replication. The total delivery vehicle cost and incentives paid are shown in Figure 6.8(a) and Figure 6.8(b), respectively. Additionally, the percentage delivery savings per replication is shown in Figure 6.8(c), whereas the average waiting time per replication is shown in Figure 6.8(d).

From Figure 6.7(a), it can be seen that there is an initial decrease in delivery cost as the `loss_aversion` parameter value is increased from 0 to 1.5. Furthermore, as the value of the `loss_aversion` parameter is increased from 1.5 to 4.5, the delivery cost seems to increase accordingly. For values greater than 4.5, there is no significant change in the total delivery vehicle cost. The total incentives paid, as shown in Figure 6.7(b), clearly decreases with an increasing `loss_aversion` parameter value. This is as a result of the threshold for classifying customers as crowd-shipping candidates becoming more restrictive, ultimately resulting in fewer instances of crowd-shipping. The savings accrued, as shown in Figure 6.7(c), seem to have an interesting relationship to the `loss_aversion` parameter value. As the `loss_aversion` parameter value is increased from 0, the savings initially increase. Once the `loss_aversion` parameter value is set greater than 3, however, the resultant savings decrease. As such, it is hypothesised that there is an intermediate range of the `loss_aversion` parameter value where the highest savings might be realised. Finally, the average waiting time, as shown in Figure 6.7(d), increases along with the `loss_aversion` parameter value.

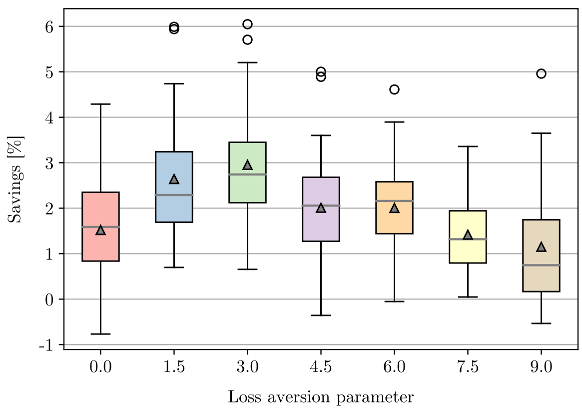
The results of Experiment 14 indicate that there is a balance to be found with respect to the value of the `loss_aversion` parameter and its associated influence on how customers are



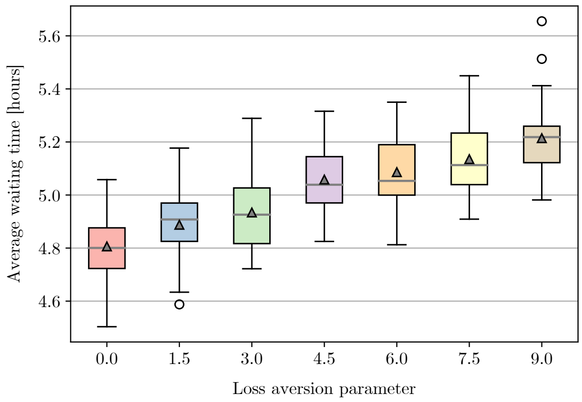
(a) *Boxplot of the total delivery vehicle cost*



(b) *Boxplot of the total incentives paid*



(c) *Boxplot of the percentage delivery savings*



(d) *Boxplot of the average waiting time*

FIGURE 6.7: *The resultant KPIs for 30 simulation replications while varying the loss aversion parameter.*

outsourced. The most liberal approach, where the `loss_aversion` parameter value equals 0, results in shorter waiting times, as all customers may be served by ODs throughout the day. The resultant savings, however, are comparatively low as a result of the liberal approach. Many customers that are served by ODs, may have been served by the regular delivery vehicles at a lower cost. As the approach becomes slightly more conservative and the customers with low cost-to-serve values are assigned to regular deliveries, a positive effect on the savings may be observed. This indicates that once the approach becomes overly conservative, negative effects are observed both with respect to savings and waiting time. In that case, the threshold is too restrictive and customer crowd-shipping is not utilised to its full capacity.

6.2 Sensitivity analysis

In this section, sensitivity analysis is performed to determine the extent to which the assumptions of the model-user influence the output of the simulation model. In particular, the parameters that describe the required model assumptions, as discussed in §4.7.1.3, are considered for the sensitivity analysis. The first set of assumptions evaluated relates to the availability of ODs, which is governed by a combination of the `OD_customer_ratio` and `base_OD_rate` parameters. Furthermore, the assumed cost of regular delivery, as governed by the `cost_per_distance` parameter, is analysed. Finally, the manner in which the OD population's value of time is modelled is analysed by evaluating the effect of changing the `minimum_VOT` parameter value. The results of the sensitivity analysis may serve as a partial validation of the simulation model, as discussed in §3.2.2. First, by observing the influence of the aforementioned assumptions and confirming it to be expected and sensible, additional confidence may be fostered in the proposed model. Additionally, the sensitivity of parameters are identified, indicating which parameters require more thorough calibration before implementation.

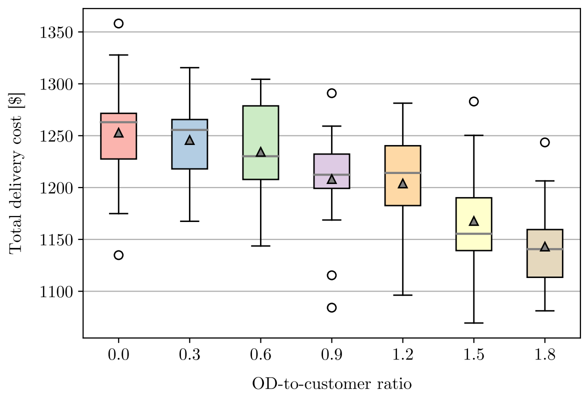
Similar to the previous section, the experiments performed in the sensitivity analysis utilise problem set R1 for 100 customers, as illustrated in Figure 6.1. All experiments are conducted over a simulated timeframe of two weeks with various simulation configurations, varying a single parameter per experiment and observing the effect on the relevant KPIs. Furthermore, experiments are conducted with multiple replications using a variable random seed, resulting in multiple independent runs. This approach is taken in an attempt to isolate, as best possible, the effect of a particular parameter and minimise the influence of other random variables in the simulation. The KPIs considered for a particular experiment may include one or more metrics of the output datasets, as discussed in §4.6. Finally, to isolate the effect of customer crowd-shipping, the number of deliveries per day and CVRP solution methodology is kept constant for all analyses. In particular, the `deliveries_per_day` is set to 2 deliveries per day, with the sweep heuristic implemented for company delivery routes.

6.2.1 OD availability

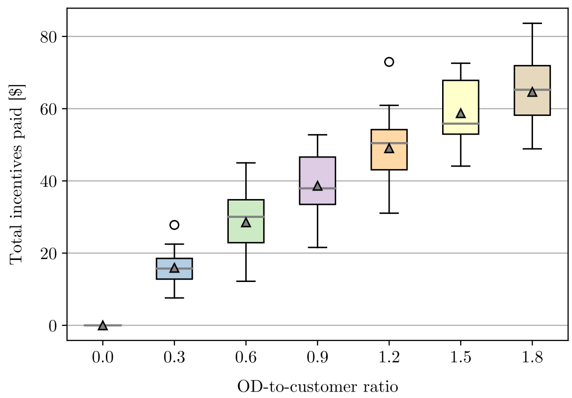
Customer crowd-shipping is dependent on the availability of ODs throughout the day. In order to determine the sensitivity of the model to variations in OD availability, a number of experiments are performed. For the proposed model, the availability of ODs for crowd-shipping is a function of both the `OD_customer_ratio` and `base_OD_rate` parameter values. As such, sensitivity analysis experiments are conducted for both of these input parameters.

The first set of experiments involve varying the `OD_customer_ratio` parameter value to analyse how sensitive the system is to the number of ODs in the system. For Experiment 15.1, the `OD_customer_ratio` parameter is varied from a value of 0 to a value of 1.8 in increments of 0.3. This is assumed to represent a reasonable range wherein the real-life ratio might fall. The accompanying parameter of concern regarding OD availability, namely the `base_OD_rate` parameter, is set to a constant value of 1 order per day. Additionally, the `minimum_VOT` parameter value is set to \$5.00 per hour. Furthermore, the default incentive scheme is applied, with the `fixed_incentive`, `variable_rate`, and `max_incentive` parameter values set to \$1.00, \$0.30 per kilometre, and \$3.00, respectively. The `cost_per_distance` and `base_order_rate` parameters are set to their respective default values of \$2.00 per kilometre and 0.25 orders per day. Finally, the `loss_aversion` parameter is set to a value of 1.

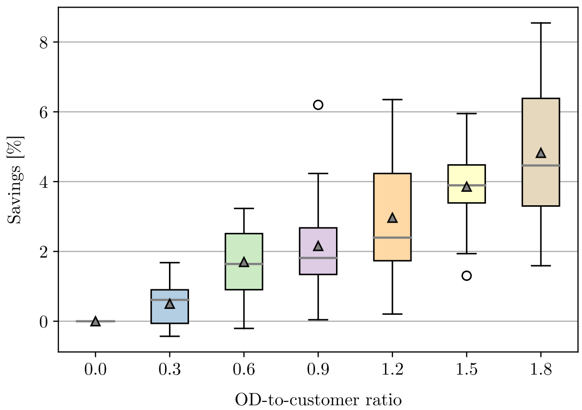
The simulation is run for a simulated time of two weeks, to observe the effect on the relevant KPIs. First, the sum of all delivery costs are considered, with respect to both the `delivery_vehicle_costs` dataset, as well as the `incentives_paid` dataset. Furthermore, the mean of all entries in the `waiting_time` dataset, as well as the value of the `percentage_savings` variable are considered. The simulation is replicated 30 times with a variable random seed resulting in independent runs. A box-plot of the relevant KPIs are shown in Figure 6.8, where each point represents the output of a single, independent replication. The total delivery vehicle cost and incentives paid are shown in Figure 6.8(a) and Figure 6.8(b), respectively. Additionally, the percentage delivery savings per replication is shown in Figure 6.8(c), whereas the average waiting time per replication is shown in Figure 6.8(d).



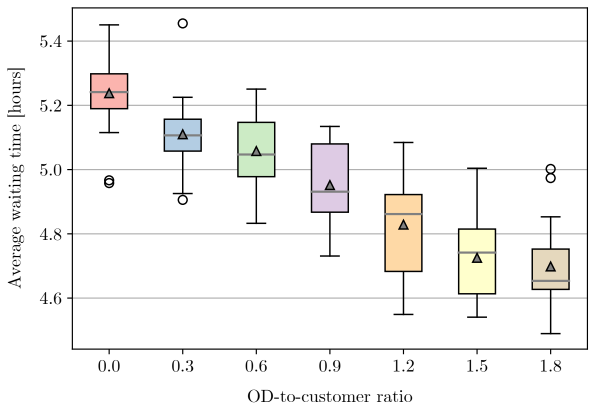
(a) *Boxplot of the total delivery vehicle cost*



(b) *Boxplot of the total incentives paid*



(c) *Boxplot of the percentage delivery savings*



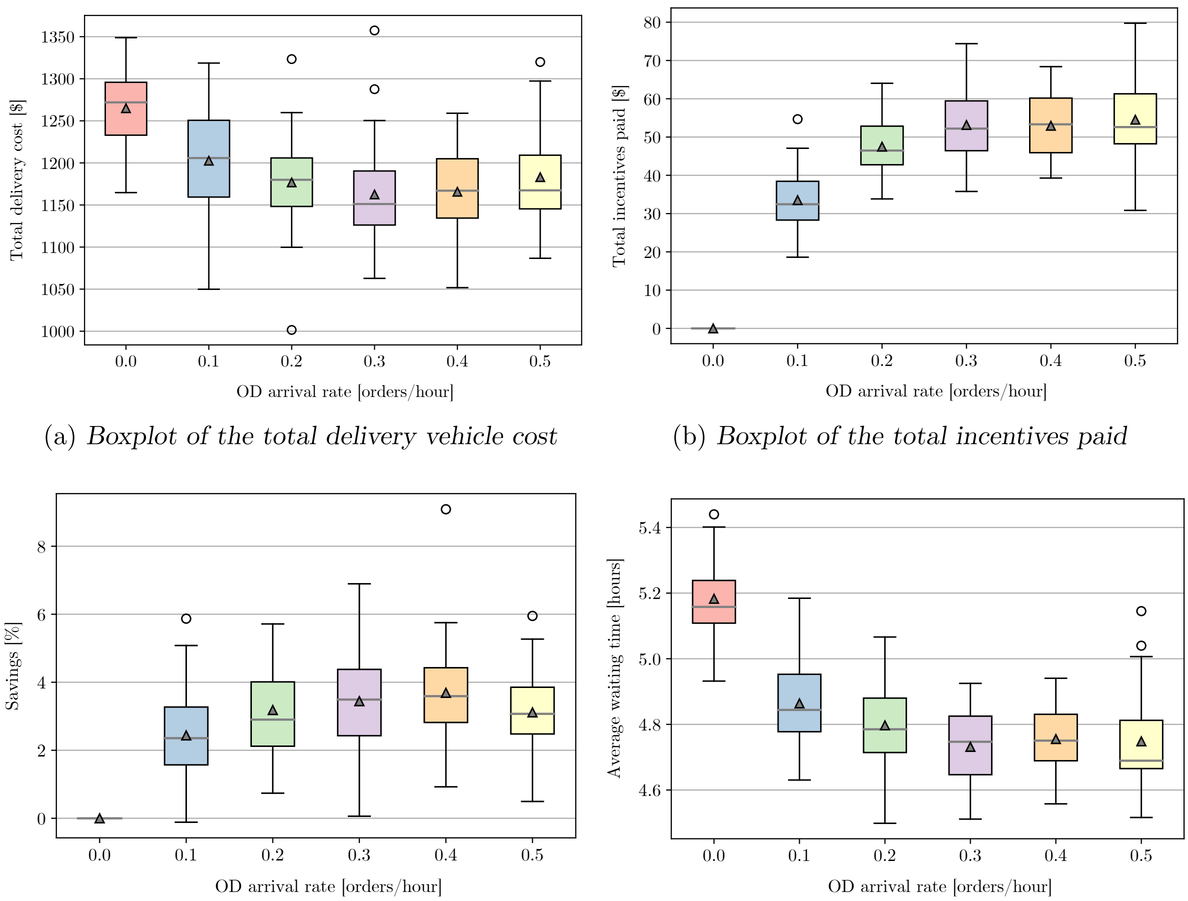
(d) *Boxplot of the average waiting time*

FIGURE 6.8: *The resultant KPIs for 30 simulation replications while varying the OD-to-customer ratio.*

As the number of ODs increases, significant effects on the output parameters are noted. First, when considering the costs displayed in Figure 6.8(a) and Figure 6.8(b), it can be seen that the cost of regular delivery decreases, while the total incentives paid increases with an increasing number of ODs. This indicates that as more ODs are available throughout the day, more successful instances of crowd-shipping are realised. Consequently, the burden on the regular delivery vehicles is decreased, while incentives are paid to ODs more regularly. The increase in the total incentives paid, however, does not overshadow the decrease in delivery cost. This is illustrated in Figure 6.8(c), where a clear increase in percentage savings is observed as the `OD_customer_ratio` parameter value is increased. Finally, it is illustrated in Figure 6.8(d) that the average waiting time decreases as more ODs are present in the system. This may once again be explained by the increased frequency at which successful crowd-shipping instances are realised. Consequently, more customers are served by ODs, rather than waiting for the regular set of deliveries.

The second set of experiments involve varying the `base_OD_rate` parameter value to analyse the effect of the frequency at which ODs visit the retailer. For Experiment 15.2, the experimental configuration of Experiment 15.1 is replicated, except for the values of the `base_OD_rate` and `OD_customer_ratio` parameter. In this experiment, the `base_OD_rate` parameter is varied from a value of 0 to a value of 0.5 in increments of 0.1. The `OD_customer_ratio` parameter, on the other hand, is kept constant at a value of 0.5.

Once again, the simulation is run for a simulated period of two weeks, replicated 30 times with a variable random seed. A box-plot of the relevant KPIs are shown in Figure 6.9, where each point represents the output of a single, independent replication. The KPIs include the total delivery vehicle cost and incentives paid, as shown in Figure 6.9(a) and Figure 6.9(b), respectively. Furthermore, the percentage delivery savings per replication is considered and shown in Figure 6.9(c), whereas the average waiting time per replication is shown in Figure 6.9(d).



(a) *Boxplot of the total delivery vehicle cost*

(b) *Boxplot of the total incentives paid*

(c) *Boxplot of the percentage delivery savings*

(d) *Boxplot of the average waiting time*

FIGURE 6.9: *The resultant KPIs for 30 simulation replications while varying the OD arrival rate.*

From Figure 6.9(a), the delivery cost seems to decrease as the `base_OD_rate` parameter value is increased from 0 to 0.2 orders per day. For the instances where the `base_OD_rate` parameter value is greater than 0.2, no clear correlation can be observed with the resultant delivery cost. On the other hand, in Figure 6.9(b), the relationship between the `base_OD_rate` parameter value and the total incentives paid is evident. Initially, ODs arrive more frequently and there is an increase in the total incentives paid. When the `base_OD_rate` parameter is set to a value greater than 0.3 orders per day, however, the total incentives paid seems to stabilise. A similar effect is noted for the savings accrued, illustrated in Figure 6.9(c). As the `base_OD_rate` parameter is increased from a value of 0 to a value of 0.2 orders per day, there is an associated increase in the percentage delivery savings. When the rate is set to values greater than 0.2, however, there is no longer an associated increase in savings. Finally, when considering the average waiting time, as illustrated in Figure 6.9(d), there seems to be an initial decrease in waiting time as the `base_OD_rate` parameter is increased from a value of 0 to 0.2. When the rate is greater than 0.3, however, no clear improvements are observed with respect to waiting time.

When considering the results of both Experiment 15.1 and Experiment 15.2, it appears that the model output is more sensitive to changes in the `OD_customer_ratio` parameter value than to changes in the `base_OD_rate` parameter value. Furthermore, as the `OD_customer_ratio` parameter value increases, a continuous effect is observed in the resulting output. On the other hand, an increase in the `base_OD_rate` parameter value only affects the output variables initially. Once the rate surpasses the rate of 0.3, its effect diminishes.

This phenomena may be understood by considering these two parameters with respect to a number of characteristics of the crowd within a crowdsourcing initiative, as described in §2.4. Particularly for customer crowd-shipping, the *largeness* of the crowd corresponds to the number of potential opportunities there are for outsourcing an order. The *diversity* of the crowd, on the other hand, involves the geographical distribution of ODs. In considering these characteristics, it is noted that the `base_OD_rate` parameter influences the largeness of the crowd, as it pertains to the frequency at which existing OD agents become available to potentially deliver orders. The geographical locations of these registered ODs, however, remain constant as the same agents visit the retailer, originating from the same home location. The `OD_customer_ratio` parameter, on the other hand, influences both the largeness and the geographical diversity of the crowd. As mentioned in §2.2.3, market penetration is critical for the success of a last-mile logistics innovation. As more OD agents are registered, the resulting geographical distribution naturally becomes more diverse, with an increasing geographical coverage. Consequently, there is a greater probability that a particular online customer that is proposed to ODs will fall into the existing path of an OD. Ultimately, more successful instances of crowd-shipping are realised.

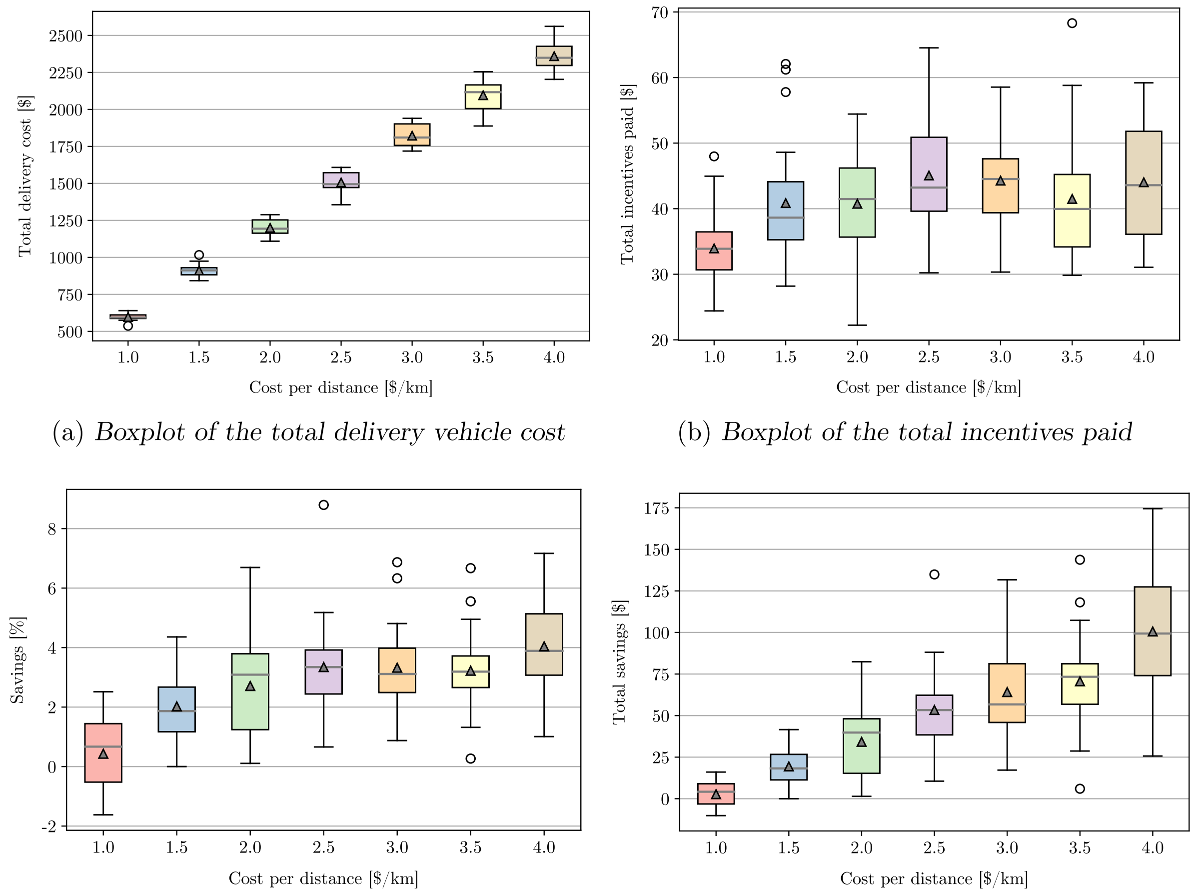
Thus, it is shown that both the largeness and geographical diversity of the crowd influences the performance of customer crowd-shipping. Moreover, there is a limit to the potential improvements resulting from increasing the largeness in isolation. Rather, both characteristics need to be considered simultaneously, with a focus on improving the frequency at which ODs become available, as well as the number of ODs in the system.

6.2.2 Validation of cost per distance

A primary motivation for the use of customer crowd-shipping is the high cost of last-mile deliveries. As such, it is necessary to consider the effectiveness of customer crowd-shipping, under varying costs of regular deliveries. In the proposed model, the cost of regular deliveries is dependent on the routing heuristic employed, as well as the cost per unit distance travelled, as described in §4.3.3.2. To evaluate the model's sensitivity to variations in the cost per distance travelled by a regular delivery, the `cost_per_distance` parameter value is varied and its influence on the relevant KPIs are analysed.

For Experiment 16, the `cost_per_distance` parameter is varied from a value of \$1.00 per kilometre to a value of \$4.00 per kilometre in increments of 0.5. The `OD_customer_ratio`, `minimum_VOT` and `base_OD_rate` parameters are set to the values of 1 OD per customer, \$5.00 per hour, and 0.6 orders per day, respectively. Furthermore, the default incentive scheme is applied, with the `fixed_incentive`, `variable_rate`, and `max_incentive` parameter values set to \$1.00, \$0.30 per kilometre, and \$3.00, respectively. Finally, the `base_order_rate` parameter is set to its default value of 0.25 orders per day.

The simulation is run for a simulated time of two weeks, to observe the effect on the relevant KPIs. First, the sum of all delivery costs are considered, with respect to both the `delivery_vehicle_costs` dataset, as well as the `incentives_paid` dataset. Furthermore, both the percentage delivery savings, as captured in the `percentage_savings` variable, as well as the sum of all entries in the `delivery_savings` dataset, are considered. The simulation is replicated 30 times with a variable random seed resulting in independent runs. A box-plot of the relevant KPIs are shown in Figure 6.10, where each point represents the output of a single, independent replication. The total delivery vehicle cost and incentives paid are shown in Figure 6.10(a) and Figure 6.10(b), respectively. Additionally, the percentage delivery savings per replication is shown in Figure 6.10(c), whereas the total delivery savings per replication is shown in Figure 6.10(d).



(a) *Boxplot of the total delivery vehicle cost*

(b) *Boxplot of the total incentives paid*

(c) *Boxplot of the percentage delivery savings*

(d) *Boxplot of the total delivery savings*

FIGURE 6.10: *The resultant KPIs for 30 simulation replications while varying the delivery cost per distance.*

The total delivery cost seems to increase linearly with an increasing cost-per distance, as illustrated in Figure 6.10(a). This result is not surprising, as the delivery vehicle costs are calculated simply as the product of the `cost_per_distance` parameter value and the distance travelled. The total incentives paid, on the other hand, indicates an interesting relationship with the `cost_per_distance` parameter, as shown in Figure 6.10(b). The total incentives paid seems to increase initially, as the `cost_per_distance` parameter is increased from a value of $1.00 per kilometre to a value of $2.50 per kilometre. This correlation diminishes, however, for instances where the `cost_per_distance` parameter is greater than $2.50 per kilometre. As the cost of serving customers with regular deliveries increases, a greater proportion of customers are classified as candidates to be outsourced. This increased number of crowd-shipping candidates explains the initial increase in the total incentives paid. At some point, however, the success of the crowd-shipping system becomes limited by the availability of ODs. As such, the correlation between the cost_per_distance parameter and the total incentives paid observed initially does not hold for larger values of the cost_per_distance parameter, as there are not enough ODs available to serve any additional online customers.

A similar trend is seen for the percentage delivery savings. As illustrated in Figure 6.10(c), the savings initially increase as the cost_per_distance parameter value is increased from $1.00 to $2.50. For values greater than $2.50, however, the percentage savings stabilises. Conversely, the total savings (*i.e.* the monetary value of the savings), as illustrated in Figure 6.10(d), seems to increase as the cost_per_distance parameter value is increased from $1.00 per kilometre to $4.00 per kilometre. Although the number of successful crowd-shipping instances become limited by the OD availability, the savings resulting from each instance becomes more valuable as the cost_per_distance parameter value is increased. As such, at a higher cost the percentage savings remain relatively constant, whereas the total delivery savings continues to increase.

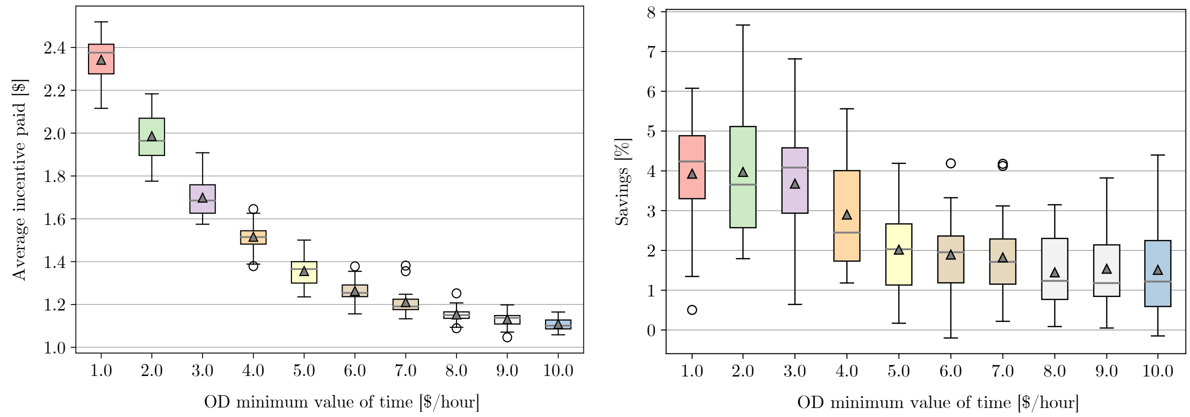
6.2.3 Validation of minimum value of time

The success of customer crowd-shipping is dependent on the willingness of ODs to perform deliveries. In the proposed model, this willingness is dependent on the OD's value of time, as described in §4.5.4. A sensitivity analysis is performed in order to determine how sensitive the model is to variations in the value of time for the OD population. The experiment involves varying the minimum_VOT parameter value and analysing its resultant effect on the relevant KPIs.

For Experiment 17, the minimum_VOT parameter value is varied from $1.00 per hour to $10.00 per hour, in increments of 1. The OD_customer_ratio and base_OD_rate parameters are set to 1 OD per customer and 0.5 orders per day, respectively. Furthermore, the default incentive scheme is applied, with the fixed_incentive, variable_rate, and max_incentive parameter values set to $1.00, $0.30 per kilometre, and $3.00, respectively. Finally, the values of the cost_per_distance and base_order_rate parameters are set to $2.00 per kilometre and 0.25 orders per day.

The simulation is run for a simulated time of two weeks, to observe the effect on the relevant KPIs. Specifically, the mean of all the entries in the incentives_paid dataset, as well as the percentage_savings variable value are considered for a given replication. The simulation is replicated 30 times with a variable random seed resulting in independent runs. A box-plot of the relevant KPIs are shown in Figure 6.11, where each point represents the output of a single replication. The average value of incentives paid per replication is shown in Figure 6.11(a), whereas the percentage delivery savings per replication is shown in Figure 6.11(b).

From the results illustrated in Figure 6.11(a), it can be seen that the average incentive decreases as the minimum_VOT parameter value is increased. As an OD's value of time increases, their perceived value of a particular deviation increases accordingly. As such, given a fixed incentive scheme, the ODs become increasingly unwilling to make large deviations. Consequently, the only offers that are accepted have small associated deviations and, accordingly, a low incentive value. Furthermore, as illustrated in Figure 6.11(b), the percentage delivery savings seems to initially decrease with an increasing minimum_VOT parameter value. The decreasing savings is explained by the fact that fewer ODs are willing to perform deliveries, resulting in fewer instances of crowd-shipping.



(a) *Boxplot of the average incentive paid* (b) *Boxplot of the percentage saving*

FIGURE 6.11: *The average incentives and percentage savings for 30 simulation replications while varying the minimum value of time parameter.*

6.3 Scenario analysis

As a final evaluation of the agent-based model for customer crowd-shipping described in Chapter 4, the performance of the model is considered for a number of scenarios. According to Law [126], the value of simulation modelling often lies in the comparison of alternatives before implementing the real-world system. As such, the aim of the scenario analysis is to gain a deeper understanding of customer crowd-shipping under various conditions. Furthermore, it aims to generate deeper insight into the workings of the agent-based model proposed in this thesis. These insights may ultimately guide the decision making when considering the implementation of a customer crowd-shipping initiative.

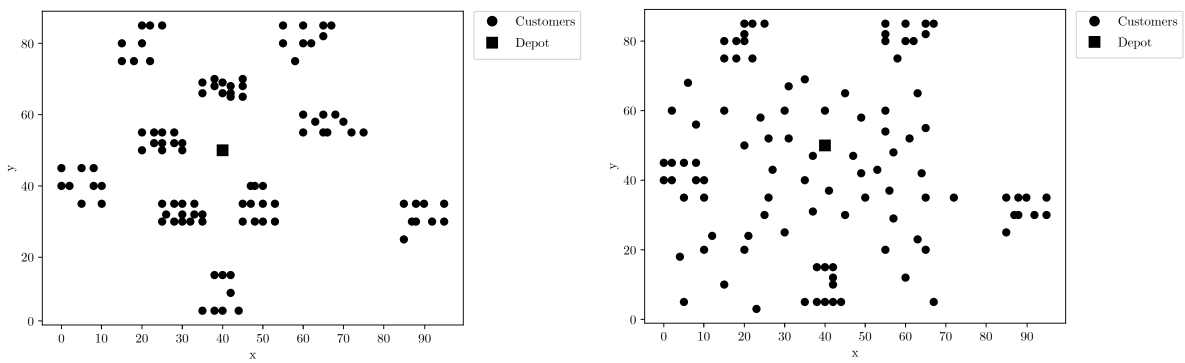
The first scenario analysis considers the spatial distribution of online customers, as well as the retailer's choice of CVRP methodology. In particular, the performance of the customer crowd-shipping model is investigated for a uniform random distribution, a clustered distribution, and a semi-clustered distribution of online customers. For each of these distributions, the performance is furthermore investigated by implementing different CVRP methodologies. In particular, the heuristic methods described in §4.3.3.2 are implemented to investigate their effect given the aforementioned customer distributions.

In the second scenario analysis, the model's performance is evaluated for six scenarios. The aim is to develop an understanding of the approach to outsourcing customers under various conditions. The conditions relate first to the maturity of a customer crowd-shipping initiative, with a distinction made between the roll-out phase and the mature phase of a system. Furthermore, different income distributions for the OD population are considered in the various scenarios. This may relate to the population wherein the crowd-shipping initiative is implemented. Finally, the effect of the `loss_aversion` parameter is investigated for each of the scenarios considered.

6.3.1 Scenario analysis I: Online customer density and routing heuristic

As mentioned in §2.2.1 and §2.2.3, customer density may have a significant impact on the effectiveness and efficiency of last-mile deliveries and should therefore be considered when implementing an initiative within this context. As such, the first scenario analysis aims to investigate the effectiveness of customer crowd-shipping for various geographical distributions of the set of online customers. The customer spatial distributions investigated correspond to the random uniform, clustered, and semi-clustered problem sets proposed by Solomon [183], as described in Appendix A. In addition to the spatial distribution of customers, the effect of the selected vehicle routing heuristic is investigated. As mentioned in §4.7.1.1, the model user may have a selection of CVRP solution methodologies. In investigating the effect of the selected solution methodology, the sweep and savings algorithms, as described in §4.3.3.2, are respectively utilised in the customer crowd-shipping model. The customer spatial distribution and the vehicle routing heuristic utilised are considered simultaneously to investigate the effectiveness of the customer crowd-shipping model.

The data utilised for the first scenario analysis correspond to the problem sets R1, C1, and RC1 for 100 customers. The uniform random problem set R1 was introduced in §6.1 and illustrated in Figure 6.1. The additional problem sets considered are illustrated in Figure 6.12. In particular, the locations of the customers and depot for the clustered problem set C1, and the semi-clustered problem set RC1 are illustrated in Figure 6.12(a) and Figure 6.12(b), respectively.



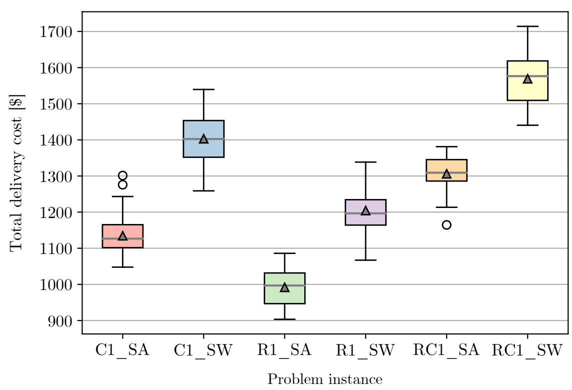
(a) *C1 problem set* (b) *RC1 problem set*

FIGURE 6.12: *The locations of customers and the depot for the C1 and RC1 problem sets from Solomon [183].*

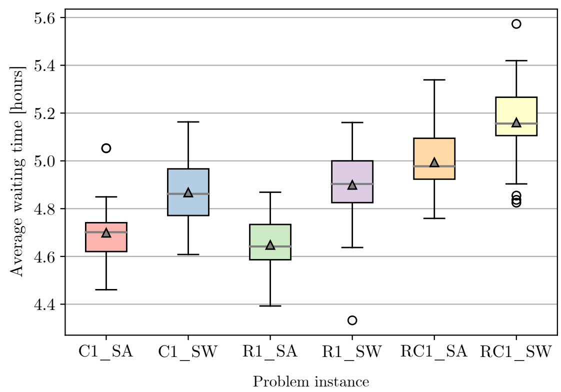
The three problem sets, namely R1, C1, and RC1, in combination with the selection of routing heuristics, namely the savings and sweep algorithms, result in six problem instances that are investigated. Problem instance `C1_SW` represents the scenario in which the clustered dataset serves as input to the model, while the sweep algorithm is utilised as the solution methodology to the CVRP. On the other hand, `C1_SA` represents the scenario in which the clustered dataset serves as input to the model, while the savings algorithm is used to solve the CVRP throughout model execution. Similarly, `R1_SW` and `R1_SA` denote the problem instances in which the dataset with uniform randomly distributed customers are used as input, with the sweep and savings algorithms respectively utilised to solve the CVRP. Finally, `RC1_SW` and `RC1_SA` denote the problem instances where the semi-cluster problem set serves as input, while the sweep and savings algorithms are utilised, respectively.

For Scenario Analysis I, the six problem instances are implemented in the customer crowd-shipping model and the relevant KPIs are observed. The availability of ODs may be described by the `OD_customer_ratio` and `base_OD_rate` parameters, which are set to the values of 1 OD per customer and 0.6 orders per day, respectively. Furthermore, the default incentive scheme is applied, with the `fixed_incentive`, `variable_rate`, and `max_incentive` parameter values set to \$1.00, \$0.30 per kilometre, and \$3.00, respectively. Additionally, the `minimum_VOT` parameter value is set to \$5.00 per hour for all scenarios. Furthermore, the `cost_per_distance` and `base_order_rate` parameters are set to their respective default values of \$2.00 per kilometre and 0.25 orders per day. Finally, the `loss_aversion` parameter is set to a value of 1.5.

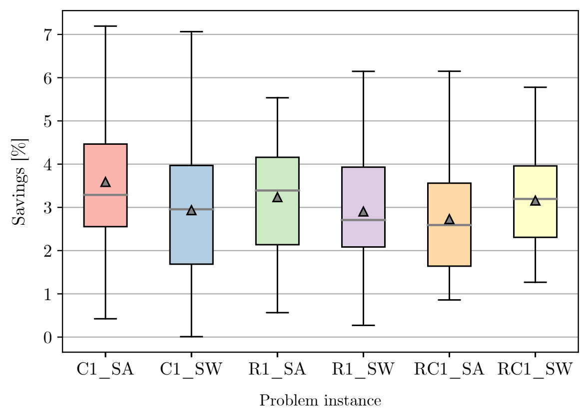
The simulation is run for a simulated time of two weeks, to observe the effect on the relevant KPIs. First, the sum of the delivery vehicle costs are considered, as captured in the `delivery_vehicle_costs` dataset. Furthermore, the average waiting time is considered, with respect to the mean of all the entries in the `waiting_time` dataset. Finally, the percentage delivery savings, as stored in the `percentage_savings` variable, is considered. The simulation is replicated 40 times with a variable random seed resulting in independent runs. A box-plot of the relevant KPIs are shown in Figure 6.13, where each point represents the output of a single, independent replication. The total delivery vehicle cost is shown in Figure 6.13(a), whereas the average waiting time per replication is shown in Figure 6.13(b). Finally, the percentage delivery savings per replication is shown in Figure 6.13(c).



(a) *Boxplot of the total delivery vehicle cost*



(b) *Boxplot of the average waiting time*



(c) *Boxplot of the percentage delivery savings*

FIGURE 6.13: *The resultant KPIs for 40 simulation replications for various problem sets in Scenario Analysis I.*

From Figure 6.13(a), the delivery vehicle costs of the various problem instances may be analysed. A summary of the delivery vehicle costs for all problem instances is tabulated in Table 6.2. The first observation is that the savings algorithm outperforms the sweep algorithm for all problem sets utilised in generating solutions with less total delivery vehicle cost. For problem instance `C1_SA`, a mean delivery vehicle cost of \$1 134 is realised, compared to the \$1 403 for problem instance `C1_SW`. Similarly, for problem instance `R1_SA` a mean delivery cost of \$991 is realised, compared to the cost of $1 204 for problem instance `R1_SW`. Finally, the delivery cost of problem instance `RC1_SA` was $1 306, compared to the cost of $1 569 for problem instance `RC1_SW`. In considering the mean, median, and standard deviation of the total delivery vehicle costs, the differences observed are quite distinct, with much better solutions obtained when utilising the savings algorithm. Furthermore, when comparing the delivery vehicle cost for different distributions, it is clear that the semi-clustered problem set RC1 incurs the highest cost, followed by C1, with R1 incurring the lowest delivery vehicle costs. This result is replicated for both the savings and sweep algorithms.

Table 6.2: *The mean, median, and standard deviation of the delivery vehicle costs obtained for all problem instances in Scenario Analysis I.*

Problem instance	C1_SA	C1_SW	R1_SA	R1_SW	RC1_SA	RC1_SW
Replications	40	40	40	40	40	40
Mean [$]	1 134	1 403	991	1 204	1 306	1 569
Median [$]	1 126	1 402	996	1 196	1 309	1 576
Standard deviation [$]	55.79	63.95	51.37	60.25	50.84	75.09

A similar result is seen in Figure 6.13(b) when considering the average customer waiting time. A summary of these results are tabulated in Table 6.3. Based on this KPI, it is observed that the savings algorithm outperforms the sweep algorithm for all customer distributions considered. Furthermore, it is observed that the problem set RC1 results in the longest average waiting time for both routing heuristics. When comparing the uniform random (R1) and clustered (C1) problem sets, however, additional analysis may be required. For problem instance `C1_SA`, an average waiting time of 4.70 hours is obtained, compared to an average waiting time of 4.65 hours for problem instance `R1_SA`. These waiting times, however, are not statistically significantly different at a significance level of 5% ($p = 0.7030$ for the Tukey post-hoc analysis). Similarly, problem instance `C1_SW` resulted in an average waiting time of 4.87 hours, whereas an average waiting time of 4.90 hours is obtained for problem instance `R1_SW`. These results, however, are also not statistically significant ($p = 0.1891$ for the Tukey post-hoc analysis). In summary, the savings algorithm results in a shorter average waiting time when compared to the sweep algorithm, for all customer distributions considered. Furthermore, the RC1 customer distribution results in a longer waiting time, compared to both the R1 and C1 customer distributions, regardless of the routing algorithm used. Finally, no distinction can be made between the average waiting times obtained for the C1 and R1 problem sets, when utilising either the sweep or savings algorithm.

Table 6.3: *The mean, median, and standard deviation of the average waiting time obtained for all problem instances in Scenario Analysis I.*

Problem instance	C1_SA	C1_SW	R1_SA	R1_SW	RC1_SA	RC1_SW
Replications	40	40	40	40	40	40
Mean [hours]	4.70	4.87	4.65	4.90	4.99	5.16
Median [hours]	4.70	4.86	4.64	4.90	4.98	5.16
Standard deviation [hours]	0.12	0.14	0.12	0.16	0.13	0.16

Finally, the percentage delivery savings obtained for each problem instance is shown in Figure 6.13(c). Additionally, the mean, median, and standard deviation of the savings obtained for each problem instance are tabulated in Table 6.4. There seems to be negligible differences across the different problem instances. Additional statistical analysis, however, may be required and an analysis of variance (ANOVA) is performed with respect to the percentage savings to analyse the result across the different problem instances. The null hypothesis of the ANOVA test states that the mean of the savings accrued for all problem instances are equal. This may be expressed as

$$H_0 : \mu_{C1\_SA} = \mu_{C1\_SW} = \mu_{R1\_SA} = \mu_{R1\_SW} = \mu_{RC1\_SA} = \mu_{RC1\_SW}.$$

The alternative hypothesis H_1 states that the mean values of the percentage delivery savings are not equal.

TABLE 6.4: *The mean, median, and standard deviation of the percentage savings obtained for all problem instances in Scenario Analysis I.*

Problem instance	C1_SA	C1_SW	R1_SA	R1_SW	RC1_SA	RC1_SW
Replications	40	40	40	40	40	40
Mean [%]	3.59	2.94	3.24	2.90	2.73	3.16
Median [%]	3.29	2.96	3.39	2.71	2.59	3.20
Standard deviation [%]	1.48	1.75	1.37	1.35	1.33	1.11

The ANOVA resulted in an F-statistic of 1.8596, which corresponds to a p-value of 0.1022. Given a significance level of $\alpha = 5\%$, there is not sufficient evidence to reject the null hypothesis. As such, there is no evidence to suggest that the percentage savings differ across the problem instances analysed in Scenario Analysis I.

The results of Scenario Analysis I provide a number of insights into the performance of the customer crowd-shipping model under various conditions. First, it is noted that the percentage delivery savings is independent of the selection in heuristic between the sweep or savings algorithms. This stands in contrast to the delivery cost and waiting time, which have been shown to be influenced by the choice of routing heuristic. As such, the results suggest that the method proposed for implementing customer crowd-shipping may increase the savings to the company regardless of the vehicle routing heuristic employed. This provides further motivation for the use of customer crowd-shipping as an additional delivery method, without necessarily having to overthrow the current routing methodology implemented by the retailer.

Furthermore, it is noted that the mean savings remain constant regardless of the geographical distribution of customers. This is contrary to the delivery cost and waiting time, as shown in Figure 6.13(a) and Figure 6.13(b), respectively, which are dependent on the geographical distribution of customers. As such, the results suggest that the company may accrue savings with the implementation of the customer crowd-shipping model irrespective of the spatial distribution of their online customers. It should be noted, however, that the geographical distribution of ODs remained constant for all experiments. In reality, the spatial distribution of ODs may coincide with that of online customers, which may influence model outcome. In summary, the outcome of Scenario Analysis I suggests that the proposed customer crowd-shipping model is robust with respect to the selected vehicle routing methodology and the geographical distribution of customers in its ability to provide solutions which may result in company savings.

6.3.2 Scenario analysis II: Crowd-shipping maturity and OD value of time distribution

For the second scenario analysis, the customer crowd-shipping model is analysed for various scenarios relating to the maturity (or popularity) of the system, as well as the income distribution of ODs in the model. Six scenarios are generated and evaluated, corresponding to two levels of maturity and three income distributions of ODs. Moreover, the `loss_aversion` parameter is varied for each scenario in order to evaluate the methodology of selecting which online customers to propose as candidates for customer crowd-shipping. A summary of the scenarios analysed is represented as a 2×3 scenario matrix in Figure 6.14. The maturity of the customer crowd-shipping system is represented on the horizontal axis and relates to the traction that the innovation has realised amongst ODs. For a model in the roll-out phase, it is assumed that there are a small number of ODs registered to participate and the `OD_customer_ratio` parameter is set to a value of 0.5. In comparison, it is assumed that for a mature system a greater number of ODs are registered and therefore participate in the system, with the `OD_customer_ratio` parameter set to a value of 1.5.

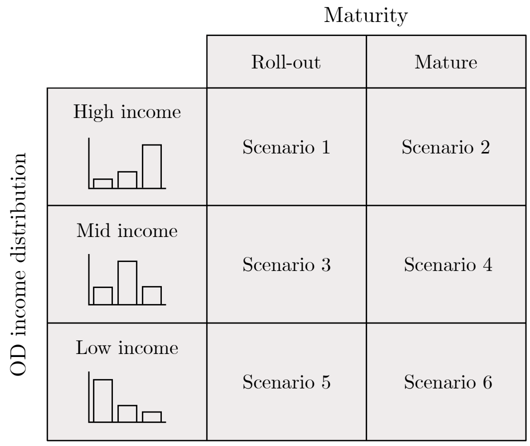


FIGURE 6.14: *An illustration of the scenarios investigated in Scenario Analysis II.*

The vertical axis, on the other hand, represents the probabilistic income distribution amongst ODs, as captured in the `income_bracket_probabilities` distribution. Particularly, the probability that an OD agent is classified into the low, middle, or high income brackets are varied for different scenarios. As discussed in §4.4.2.2, these probabilities ultimately influence the overall population income distribution and, consequently, the value of time distribution for the population of ODs. The first distribution emulates a scenario where most ODs may be classified into the high income bracket, with $p_{low} = 0.1$, $p_{mid} = 0.3$, and $p_{high} = 0.6$. The second distribution emulates a scenario where most ODs fall into the middle income bracket, with $p_{low} = 0.25$, $p_{mid} = 0.5$, and $p_{high} = 0.25$. Finally, the third distribution emulates a scenario with ODs primarily from the low income bracket, with $p_{low} = 0.6$, $p_{mid} = 0.3$, and $p_{high} = 0.1$. Moreover, in order to isolate the effect of income probabilities on the OD population's value of time distributions, the `minimum_VOT` parameter value is set to \$5.00 per hour for all scenarios.

Finally, the `loss_aversion` parameter value is varied for each scenario to evaluate its influence on the model under various conditions. This provides an indication of how varying the outsourcing approach (*i.e.* from being more liberal or more conservative) influences the model performance for each scenario.

For Scenario Analysis II, the six scenarios are implemented in the customer crowd-shipping model and the relevant KPIs are observed. For each scenario, the `loss_aversion` parameter, depicted as β, is varied from a liberal value of 0 to a conservative value of 6 in increments of 1.5. This ultimately results in 30 simulation configurations — corresponding to five `loss_aversion` parameter values for each of the six scenarios. In all cases, the randomly distributed problem set R1 is used to describe the geographical locations of customers. Furthermore, the savings algorithm is selected as the CVRP solution methodology. Although the `OD_customer_ratio` parameter value varies for the given scenario maturity, the `base_OD_rate` parameter remains constant at a value of 0.6 orders per day. Furthermore, the default incentive scheme is applied, with the `fixed_incentive`, `variable_rate`, and `max_incentive` parameter values set to \$1.00, \$0.30 per kilometre, and \$3.00, respectively. Finally, the `cost_per_distance` and `base_order_rate` parameters are set to their respective default values of \$2.00 per kilometre and 0.25 orders per day.

The simulation is run for a simulated time of two weeks, and the effect on the relevant KPIs are observed. For this analysis, the primary KPIs of concern include the percentage savings and the average waiting time. As such, the mean of all the entries in the `waiting_time` dataset, as well as the percentage savings, as stored in the `percentage_savings` variable, are captured for each simulation run. For each experimental configuration, the simulation is replicated 40 times with a variable random seed resulting in independent runs.

As discussed in §3.2.5, the stochastic nature of simulation modelling calls for careful interpretation of the output. As such, the output of the 40 replications are utilised to construct confidence intervals for the true mean of both the delivery savings and the average waiting time for each simulation configuration. This provides an approximation for the true mean of the variables for each configuration. An α-value of 0.05 is selected, corresponding to confidence intervals with 95% coverage.

The output relating to the percentage delivery savings is considered in Table 6.5, detailing the sample mean $\bar{x}$, standard deviation s, as well as the confidence interval for each simulation configuration. Similarly, the average waiting time for all configurations are tabulated in Table 6.6, detailing the mean $\bar{x}$, standard deviation s, and confidence interval for a particular simulation configuration. Furthermore, to provide a graphical representation of the results, the point estimators (*i.e.* the mean of the 40 replications) of the percentage savings and average waiting time for each configuration are plotted in Figure 6.15.

TABLE 6.5: *The mean $\bar{x}$, standard deviation s, and 95% confidence interval for the percentage savings obtained for the roll-out and mature phases of customer crowd-shipping with various income distributions, and a varying loss aversion parameter β.*

		Roll-out phase			Mature phase		
	β	$\bar{x}$ [%]	s [%]	95% CI	$\bar{x}$ [%]	s [%]	95% CI
High income	0.0	0.921	0.808	[0.662, 1.179]	3.021	1.482	[2.547, 3.495]
	1.5	1.378	0.756	[1.136, 1.620]	3.661	1.284	[3.251, 4.072]
	3.0	1.394	0.924	[1.098, 1.689]	3.559	1.054	[3.222, 3.896]
	4.5	1.157	0.770	[0.911, 1.404]	2.755	1.320	[2.333, 3.177]
	6.0	0.630	0.659	[0.420, 0.841]	1.889	1.129	[1.528, 2.250]
Mid income	0.0	1.282	1.173	[0.907, 1.657]	3.676	1.451	[3.212, 4.140]
	1.5	1.804	0.966	[1.495, 2.113]	4.765	2.064	[4.105, 5.425]
	3.0	1.683	1.009	[1.360, 2.005]	4.398	1.391	[3.953, 4.843]
	4.5	1.471	1.015	[1.146, 1.795]	3.613	1.492	[3.136, 4.090]
	6.0	0.928	0.801	[0.672, 1.184]	2.706	1.253	[2.305, 3.107]
Low income	0.0	1.530	1.272	[1.123, 1.937]	4.011	1.472	[3.540, 4.481]
	1.5	1.900	0.938	[1.600, 2.200]	5.142	1.771	[4.576, 5.708]
	3.0	2.385	1.107	[2.032, 2.739]	5.245	1.716	[4.696, 5.794]
	4.5	1.882	1.268	[1.477, 2.288]	4.261	1.541	[3.768, 4.754]
	6.0	1.632	0.954	[1.327, 1.937]	3.667	1.636	[3.144, 4.190]

TABLE 6.6: *The mean $\bar{x}$, standard deviation s, and 95% confidence interval for the average waiting time obtained for the roll-out and mature phases of customer crowd-shipping with various income distributions, and a varying level of the loss aversion parameter β.*

	β	Roll-out phase $\bar{x}$ [hours]	s [hours]	95% CI	Mature phase $\bar{x}$ [hours]	s [hours]	95% CI
High income	0.0	4.923	0.103	[4.890, 4.956]	4.618	0.132	[4.575, 4.660]
	1.5	4.931	0.122	[4.892, 4.970]	4.662	0.093	[4.632, 4.691]
	3.0	4.959	0.118	[4.921, 4.997]	4.762	0.111	[4.726, 4.797]
	4.5	5.019	0.133	[4.976, 5.061]	4.865	0.109	[4.830, 4.900]
	6.0	5.066	0.108	[5.032, 5.101]	4.944	0.120	[4.906, 4.983]
Mid income	0.0	4.867	0.114	[4.830, 4.903]	4.518	0.142	[4.472, 4.563]
	1.5	4.949	0.141	[4.904, 4.994]	4.576	0.140	[4.531, 4.621]
	3.0	4.929	0.099	[4.897, 4.961]	4.713	0.155	[4.663, 4.763]
	4.5	4.984	0.090	[4.955, 5.013]	4.791	0.110	[4.756, 4.826]
	6.0	5.025	0.129	[4.984, 5.066]	4.888	0.123	[4.849, 4.928]
Low income	0.0	4.771	0.109	[4.736, 4.806]	4.458	0.119	[4.420, 4.496]
	1.5	4.854	0.131	[4.812, 4.896]	4.491	0.117	[4.454, 4.529]
	3.0	4.865	0.106	[4.831, 4.899]	4.583	0.126	[4.542, 4.623]
	4.5	4.901	0.129	[4.859, 4.942]	4.726	0.149	[4.679, 4.774]
	6.0	5.008	0.116	[4.970, 5.045]	4.814	0.129	[4.773, 4.855]

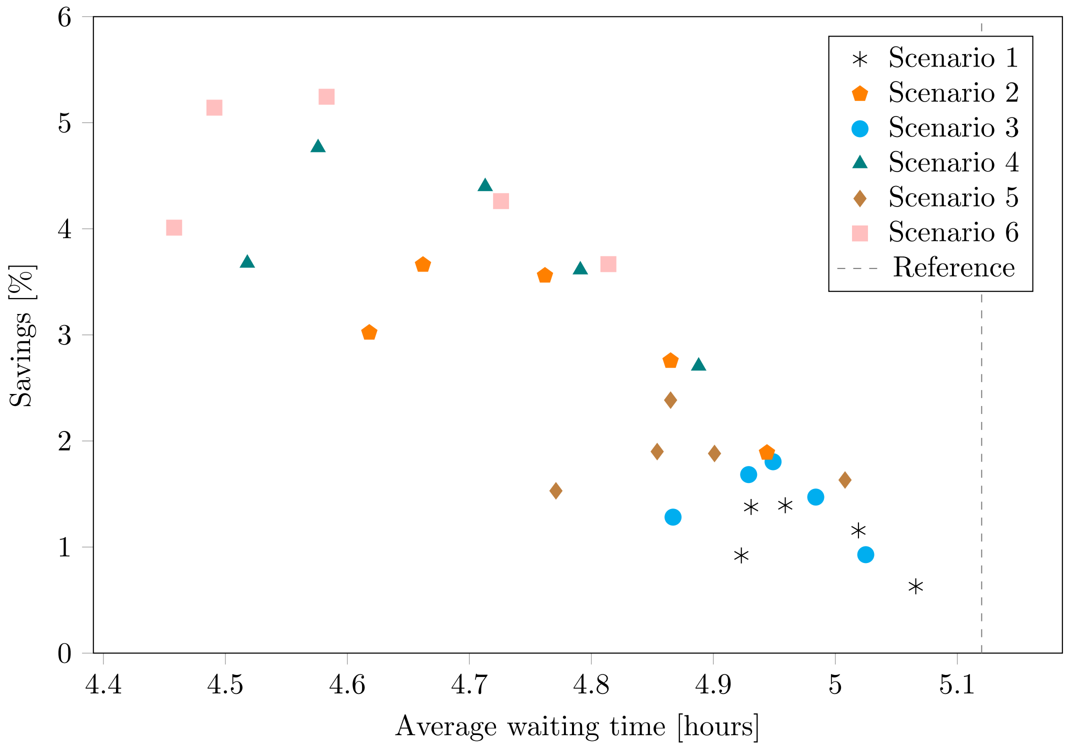


FIGURE 6.15: *The point estimates of the average waiting time and percentage savings for the six scenarios with varying loss aversion parameter values as considered in Scenario Analysis II.*

The point estimates plotted in Figure 6.15 provide a good starting point in interpreting the results of the scenario analysis. It can be seen that there are major variation across different scenarios, but also within each scenario as the `loss_aversion` parameter value is varied. To provide context of the improvement in average waiting time, a reference line represents the mean average waiting time when customer crowd-shipping is not utilised and the savings algorithm is employed, as calculated in §6.1.1 and captured in Table 6.1. It is observed that the customer crowd-shipping initiative results in improvements with respect to the waiting time and delivery cost for all configurations evaluated. The results may further be analysed by considering the variations in the system maturity, the income distribution, as well as the loss aversion parameter employed.

The first variation considered relates to the comparison of roll-out and mature scenarios. It is clear that the scenarios in a mature phase (*i.e.* Scenarios 2, 4, and 6) perform substantially better compared to their counterparts in a roll-out phase (*i.e.* Scenarios 1, 3, and 5). In particular, when comparing Scenario 1 to Scenario 2, it is evident that greater savings are consistently realised in Scenario 2, irrespective of the value of the loss aversion parameter. Additionally, the average waiting time observed in Scenario 2 is generally shorter, compared to that observed in Scenario 1. A choice of $\beta = 6$ in Scenario 2, however, results in a longer waiting time than some of the configurations in Scenario 1. A similar trend is observed when comparing the results of Scenario 3 and Scenario 4. Greater savings are consistently realised in Scenario 4, when comparing across all choices of the loss aversion parameter. An accompanying shorter waiting is generally observed in Scenario 4, however, a choice of $\beta = 6$ in Scenario 4 results in a longer waiting time in comparison to the shortest waiting time observed in Scenario 3. Finally, these results are replicated when comparing the outcomes of Scenario 5 and Scenario 6 — greater savings with a shorter average waiting time are generally realised in the mature scenario. Once again, a choice of $\beta = 6$ in Scenario 4 results in its worse performance with respect to waiting time, and is outperformed by the shortest waiting time in Scenario 5. More generally, it is clear that the maturity (or popularity) has a major influence on the performance of the system, with more mature systems significantly outperforming those in a roll-out phase.

The effect of the income distribution may be seen in Figure 6.15, although it is not as acute as the influence of system maturity. First, the effect of the income distribution across systems in their roll-out phase (*i.e.* Scenarios 1, 3, and 5) are considered. It appears that the best performances are achieved in Scenario 5, while the worst performances are observed for Scenario 1. This indicates that when there is a larger proportion of ODs from a low income bracket, more successful instances of crowd-shipping are realised. This may be due to a lower threshold for their willingness to deliver an order, compared to the incentive offered. Given that the incentive scheme remains fixed for all scenarios, the reduced threshold leads to greater delivery savings, as well as a shorter average waiting time. This effect is replicated for the scenarios in the mature phase (Scenarios 2, 4, and 6). It can be seen that the income distribution has a more exaggerated effect on the scenarios in a mature phase compared to those in the roll-out phase. In a mature system, the best performances are realised in Scenario 6 (*i.e.* a high proportion of ODs in the low income bracket), whereas the worst performances resulted from Scenario 2 (*i.e.* a high proportion of ODs in the high income bracket).

These results indicate that the value individual's place on their time and, consequently, their WTA may have a significant influence on the success of a customer crowd-shipping initiative. When there is a significant proportion of ODs for which the incentives offered are of sufficient value, significant savings may be realised as well as a reduced waiting time. As their perceived value of time increases, however, the improvements made from the use of crowd-shipping diminishes. It is noted that income may not be the only factor that influences an individual's value of time. Moreover, it is noted that an individual's willingness to participate in an initiative of this sort may be influenced by additional factors, such as shifting consumer values, environmental consciousness, and the movement towards community-oriented living, as mentioned in §2.3.4.1. The results still carry significance, however, and may serve as a basis for understanding the manner in which initiatives such as customer crowd-shipping are supported.

Finally, the effect of the loss aversion parameter is analysed for all scenarios considered. A similar trend is observed for all scenarios when varying the loss aversion parameter. For the most liberal approach to outsourcing customers ($\beta = 0$), the shortest average waiting time is observed, with a positive associated percentage delivery savings. As the approach becomes moderately conservative ($1.5 \leq \beta \leq 3$), the average waiting time increases. An improvement is observed, however, for the percentage delivery savings achieved. In all cases, the highest percentage delivery savings is achieved for a `loss_aversion` parameter value of either 1.5 or 3. As such, it is noted that the performance with respect to waiting time is sacrificed for a greater percentage savings by moving from a liberal to a moderate approach. As the approach becomes increasingly conservative ($\beta > 3$), there is an associated increase in average waiting time as well as a reduction of the percentage delivery savings. This implies that, for all scenarios, a highly conservative approach results in the deterioration of the customer crowd-shipping performance.

Furthermore, the effect of changes in the `loss_aversion` parameter value seems to be more pronounced in mature systems as compared to systems in the roll-out phase. Moreover, it is observed that a poor choice of the loss aversion parameter may, even in cases where external conditions are favourable, result in poor performance of a customer crowd-shipping initiative. This may be exemplified by comparing the performance for Scenario 4 and Scenario 6. For Scenario 6, in the case where the `loss_aversion` parameter is set to a value of 4.5, a mean average waiting time of 4.726 hours is observed, with a mean percentage delivery savings of 4.261%. Even though the conditions in Scenario 4 are less favourable to the system (with respect to the OD income distribution), an appropriate selection of the `loss_aversion` parameter may result in better performance than realised in Scenario 6. In particular, for Scenario 4, a selection of 1.5 and 3 for the `loss_aversion` parameter value outperforms the cases in Scenario 6 where a `loss_aversion` parameter value of 4.5 and 6 are selected. A similar result is observed across various scenarios, where the appropriate choice of the loss aversion parameter results in better performance compared to a scenario that is seemingly more favourable for customer crowd-shipping. As such, it is noted that the approach to selecting customers to outsource has a significant influence on the performance of a customer crowd-shipping initiative.

In order to further evaluate the different scenarios, the configurations that resulted in the best performance for each of the six scenarios are considered. Particularly, the non-dominated solutions for a particular scenario, with respect to percentage delivery savings and waiting time, are considered. To illustrate these solutions, the point estimators (*i.e.* the mean of the 40 replications) of the percentage savings and average waiting time for each non-dominated solution are plotted in Figure 6.16. Additionally, the confidence intervals of both variables (as tabulated in Table 6.5 and Table 6.6) are shown along with each point estimator. This provides an indication of the separation for the different configurations.

For each scenario depicted in Figure 6.16, the non-dominated solutions range from a liberal approach (*i.e.* focussed on reducing waiting time) to a more conservative approach (*i.e.* focussed on reducing delivery cost). This is ultimately based on the selected `loss_aversion` parameter value, which defines how liberally customers are proposed as candidates for customer crowd-shipping. It is observed, however, that for all non-dominated instances, the `loss_aversion` parameter value is less than 3. This implies that the liberal to moderate approaches to selecting customers for outsourcing resulted in the best performance of the crowd-shipping initiative.

A retailer's choice of the `loss_aversion` parameter value may be influenced by the make up of their OD profiles as well as the relative importance of the respective KPIs. For the scenarios

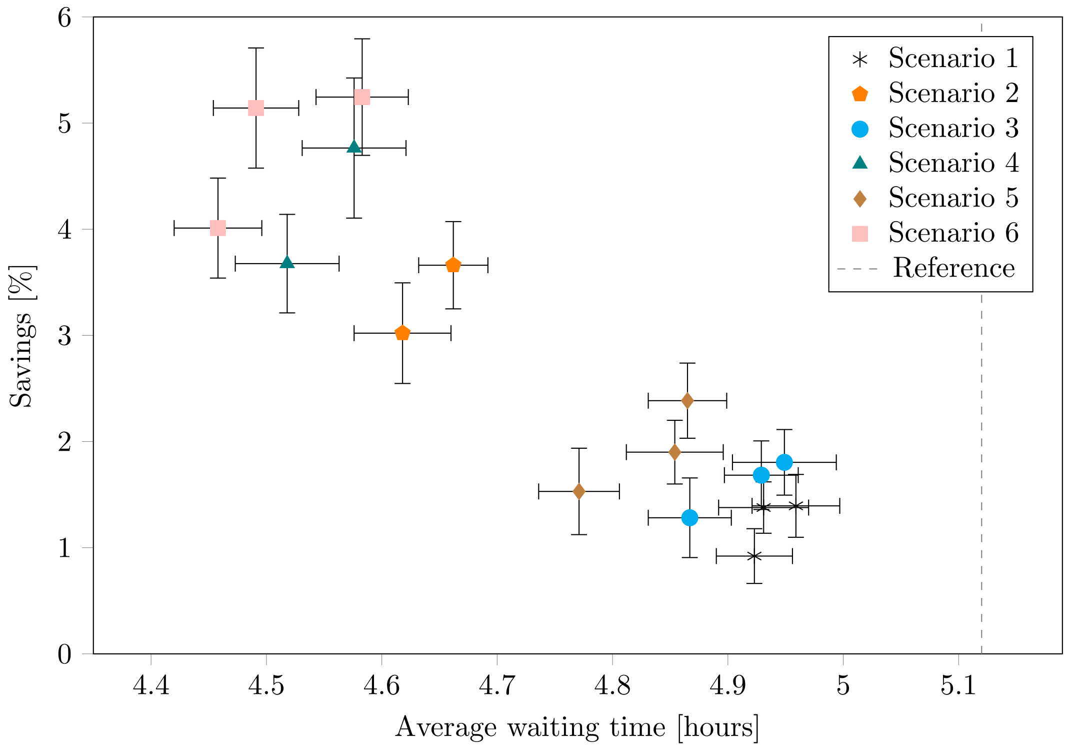


FIGURE 6.16: *The point estimators and 95% confidence intervals for the percentage savings and average waiting time of the non-dominated solutions for each scenario in Scenario Analysis II.*

in the roll-out phase (*i.e.* Scenarios 1, 3, and 5), for instance, a `loss_aversion` parameter value of 0 might be most beneficial to the long-term success of the customer crowd-shipping system. Given that the average waiting time may be considered as a customer-facing KPI (*i.e.* its performance influences customer satisfaction directly), the retailer might consider a liberal approach to customer crowd-shipping during the roll-out phase. Although such an approach may not result in the most savings, the potential for growth of the customer crowd-shipping system should be considered, rather than short-term savings. Particularly, as customer satisfaction improves through the use of customer crowd-shipping due to the reduced waiting time, the concept may become increasingly attractive and gain traction. Such a focus on improved customer experience may lead the way towards a mature customer crowd-shipping system. Once maturity is reached, the objectives of the retailer may shift from customer acquisition to a more conservative approach, potentially focussing on delivery savings. In these mature scenarios (*i.e.* Scenarios 2, 4, and 6), a more conservative choice of the `loss_aversion` parameter values may be more appropriate.

6.4 Subject matter expert validation

An additional measure to attain credibility and foster confidence in a simulation model includes face validation by key project personnel or industry experts [126]. To this end, a number of subject matter experts with experience in the fields of ABM, urban logistics, sustainable road freight, the collaborative economy, customer behaviour, and economics were consulted. The proposed agent-based model was presented to a panel comprising three research associates in the Centre for Sustainable Road Freight at Heriot-Watt University. Dr Dhanan Utomo [196] holds a PhD in management science from Lancaster University Management School and currently works in developing simulation models in the realms of logistics and supply chains. In particular, Dr Utomo focusses on ABM and has an interest in developing methodologies to elicit decision making rules of real-world actors. Furthermore, Dr Nadia Taou [186] holds a PhD in Computer Science from Heriot-Watt University. Her focus pertains to using data science and machine learning techniques for research in logistics and economics. Finally, Pratyush Dadhich [62] is a PhD candidate in logistics and supply chain management at Heriot-Watt University, and has been a research associate at the Centre for Sustainable Road Freight since 2013. He has experience in analysing supply chain networks in identifying horizontal collaboration opportunities, in addition to experience in utilising ABM to simulate urban deliveries, and the associated customer behaviour.

The proposed agent-based model for customer crowd-shipping was presented to the subject matter experts in a structured manner. This included the detailing of the model workings, assumptions, inclusions, exclusions and solution methodologies used in the model. In particular, emphasis was placed on describing the incentive scheme utilised, the approach towards selecting online customers as candidates for outsourcing, as well as the decision making of ODs. This was done both from a model design perspective, as well as based on the verification tests with the animation output of the model. Additionally, the results obtained from a number of experiments were presented. This enabled the panel to ascertain how realistically and appropriately the various elements of customer crowd-shipping had been incorporated, while additionally confirming the validity of the results obtained.

The presentation was received well by the panel, with an overwhelming positive response to the modelling approach. The subject matters experts agreed with the general assumptions made, including the manner in which the traditional last-mile logistics system is modelled, as well as the theoretical implementation of customer crowd-shipping. They agreed that although additional information would be required for modelling a particular implementation of customer crowd-shipping, the proposed model serves as a useful and realistic concept demonstrator. Moreover, the agent-based approach was complimented, emphasising the adequate modelling of OD decision making. In particular, the autonomous nature of ODs was encouraged and complimented.

Additionally, a number of suggestions were made by the panel, relating to potential future research that may follow from this study. A number of considerations were mentioned pertaining to the implementation of a real-world customer crowd-shipping initiative. First, it was suggested that the cost structure of dedicated delivery vehicles be extended to include a fixed cost, and wage rate in addition to the distance-based rate employed in the model. Furthermore, the panel was of the opinion that ODs may be likely to carry multiple orders during a single trip. As such, it was suggested that the capacity of OD agents be extended, enabling them to perform multiple deliveries. Finally, it was suggested that the type of delivery vehicles used by both the dedicated fleet and ODs be specified for more precise considerations with respect to the vehicle capacity and the carbon emissions of the system.

6.5 Chapter summary

In this chapter, the agent-based model proposed in Chapter 4 and verified in Chapter 5, was validated and evaluated. The aim of this chapter included generating deeper insight into the model and customer crowd-shipping in general, by evaluating the model output under a number of scenarios. Furthermore, confidence in the model was fostered, reasonable values for the various input parameters were identified, and the performance of customer crowd-shipping was evaluated under various conditions. This was achieved through the use of parameter variation, sensitivity analysis, scenario analysis, and validation from subject matter experts. In §6.1, the parameter variation experiments were documented, and insight was gained into the effect of the decisions made by the retailer. Similarly, in §6.2, the model's sensitivity to a number of assumptions was analysed. Furthermore, a number of scenarios were proposed in §6.3, and the model output for these scenarios were analysed and discussed. Finally, in §6.4, the face validation performed by subject matter experts was documented.

CHAPTER 7

Conclusion

Contents

A summary of the contents of this thesis is provided in §7.1. This is followed by an appraisal of the contributions made in the thesis, in §7.2. The chapter is concluded in §7.3 with a number of suggestions with respect to potential future work that may stem from the research conducted in this thesis.

7.1 Thesis summary

In addition to this final chapter, the thesis comprised six chapters aimed at addressing the problem statement and fulfilling the stated objectives. In the introductory chapter of this thesis, Chapter 1, a brief background of the problem is provided. In particular, customer crowd-shipping is introduced as an innovation to improve the last-mile logistics systems of omni-channel retailers. Furthermore, a number of real-world implementations of customer crowd-shipping was mentioned, in addition to a number of existing quantitative models capturing this phenomenon in the literature. ABM was introduced as an approach towards modelling the customer crowd-shipping initiative, with reference to its bottom-up approach and ability to model autonomous agents. Finally, the scope and objectives of the study were detailed, along with a brief outline of the thesis organisation.

In Chapter 2, a literature review was presented detailing the various background considerations for customer crowd-shipping, in partial fulfilment of Objective I. This chapter includes a background of the e-commerce industry, as well as an in-depth discussion of the associated last-mile logistics, in fulfilment of Objective I(a). Thereafter, the collaborative economy was discussed in fulfilment of Objective I(b). In particular, the discussion considered the shift in consumer behaviour towards more collaborative lifestyles, the various existing business models within this realm, as well as the forces driving these modern business ventures. Finally, an in-depth discussion on crowdsourcing and, in particular, crowd logistics was provided in fulfilment of Objective I(c). A framework for crowdsourcing models was provided, and the field of crowd logistics was discussed with reference to this framework. Finally, a number of real-world implementations of crowd logistics were discussed to gain a better understanding of the current landscape within the field.

In Chapter 3, a further review was conducted focussing on various modelling aspects, techniques, and approaches in the existing literature, in final fulfilment of Objective I. This chapter was primarily focussed on providing the background information necessary to understand the development of the proposed agent-based model and its various constituent elements. First, the domain of VRPs was discussed, in fulfilment of Objective I(d). Particular focus was placed on the classical CVRP, detailing a number of existing solution methodologies found in the literature. Thereafter, an overview of computer simulation modelling was provided, in fulfilment of Objective I(e). This included discussions on the classification of simulation models, the advantages of simulation modelling, and the various modelling paradigms available, with a particular focus on ABM. Thereafter, an overview of microeconomic theory was provided, with specific reference to consumer behaviour, in fulfilment of Objective I(f). In particular, the value that consumers place on their travel time, their willingness to trade travel time for money, as well as the modelling approaches thereof, were discussed. Finally, existing models of crowd logistic initiatives were discussed, in fulfilment of Objective I(g), in providing context of the existing quantitative modelling perspectives in the literature relating to customer crowd-shipping.

In Chapter 4, the development and implementation of an agent-based model of customer crowd-shipping was discussed, in fulfilment of Objectives II and III. The development of the model was described in three phases. In the first phase, the modelling of a traditional e-commerce retailer serving a number of online customers from a single depot was detailed, in fulfilment of Objective II(a). This included a discussion on the selection of input data pertaining to online customers, the configuration of the last-mile logistics system, as well as the various CVRP solution methodologies implemented in the proposed model. In the second phase, the general behaviour and decision making of ODs are described in partial fulfilment of Objective II(b). In particular, this phase pertains to the value that ODs associate with their travel time. In the third phase of model development, an integrated approach to customer crowd-shipping was proposed, in final fulfilment of Objective II(b) and in fulfilment of Objective II(c). This phase detailed the development of a cost-to-serve algorithm, in fulfilment of Objective III(d), as well as the proposed incentive scheme utilised, the proposed method of selecting customers for outsourcing, and the decision making of ODs. Performance measures were established by introducing and discussing a number of KPIs for the customer crowd-shipping model, in partial fulfilment of Objective III(f). In addition to this, the development of a GUI was discussed, governing both the implementation and interpretation of simulation runs. This included discussions on the user's selection of input data pertaining to online customers, delivery vehicles, and their choice of CVRP solution methodology, in fulfilment of Objectives III(a) and (b), respectively. Furthermore, the details of the user-defined incentive scheme, as well as the user-defined approach for selecting customers to propose as crowd-shipping candidates to ODs were discussed, in fulfilment of Objectives III(c) and (e), respectively. Finally, the GUI's capacity to graphically illustrate the execution of a simulation run and the associated performance of the customer crowd-shipping model, were discussed in final fulfilment of Objective III(f).

Chapter 5 comprised a discussion on the verification of the agent-based model proposed in this thesis, in fulfilment of Objective IV. First, a number of verification techniques was discussed. Thereafter, the systematic verification of a number of key modelling elements were considered. These include the customer order frequency, the planning and execution of last-mile deliveries, the proposed cost-to-serve algorithm, the behaviour of ODs, as well as the calculation of delivery savings. The documentation of verification tests was structured in accordance to the sequential and iterative verification approach applied throughout model development, whereby a new layer of complexity was only added once the current model was sufficiently verified.

In Chapter 6, the evaluation of the agent-based model for customer crowd-shipping was discussed. First, a number of parameter variations and sensitivity analyses were conducted, and the model outcome was analysed, in partial fulfilment of Objective V. For the parameter variations, the user-defined variables were investigated, and their influence on the model performance were analysed. For the sensitivity analysis, on the other hand, the assumptions made about the customer crowd-shipping system were investigated, and their associated influence on the model performance were analysed. Finally, a scenario analysis was conducted, and the performance of the customer crowd-shipping model under a number of conditions were discussed, in fulfilment of Objective VI. This included an investigation on the effect of online customer distribution in conjunction with various implemented vehicle routing heuristics. Additionally, the maturity of the crowd-shipping initiative was investigated in conjunction with varying income distributions. Finally, the proposed model was subjected to face validation by subject matter experts, who validated the model as adequate, in final fulfilment of Objective V.

The present chapter serves as the conclusion of this thesis and includes, in fulfilment of Objective VII, sensible follow-up work which may stem from the research performed in this thesis.

7.2 Thesis contributions

The main contributions of this thesis towards furthering the understanding and modelling approach of customer crowd-shipping initiatives are presented in this section. The contributions relate to the understanding of customer crowd-shipping as it is applied in the real-world, as well as the theoretical models hitherto proposed to represent this innovation. Moreover, additional contributions are made by the incorporation of this improved understanding of customer crowd-shipping into the modelling approach, furthering the capacity to capture the real-world workings of customer crowd-shipping adequately.

Contribution 1 *A synthesis of the literature pertaining to customer crowd-shipping.*

The customer crowd-shipping initiative is a relatively new innovation both in the literature and in practice. In this thesis, an outline of the relevant literature pertaining to the fundamental building blocks of such an initiative was provided, in addition to a comprehensive synthesis of the quantitative work hitherto performed to model a customer crowd-shipping initiative. The former relates first to the e-commerce industry and the associated field of last-mile logistics, wherein customer crowd-shipping initiatives may be embedded. Furthermore, the growing movement of the collaborative economy was introduced, detailing such collaborative business models targetted at being more efficient, effective, and sustainable. In particular, the field of crowdsourcing was discussed in depth, with reference to a framework of understanding such business models. Finally, the culmination of these constituent fields, namely crowd logistics, was discussed. These fields provide the fundamental building blocks required to understand both the origins, the driving forces, as well as the potential future of customer crowd-shipping. The synthesis of the literature provides a basis for further research into the field, while simultaneously serving as a guide for those considering the implementation of such a system in the real world.

An additional contribution is made through the documentation of a comprehensive survey of existing crowd logistics models pertaining to customer crowd-shipping. This synthesis provides the lay of the land with respect to models derived for depicting the instance where a company utilises their in-store customers to deliver orders to online customers. In addition to an in-depth discussion of each proposed model, a set of notable aspects required for the modelling of a customer crowd-shipping initiative was identified. This includes the arrival of online orders and in-store customers, the incentive scheme utilised, as well as the behaviour and decision-making process of ODs. The various existing approaches to modelling these critical aspects, as documented in the literature, were evaluated and compared.

Contribution 2 *The design and development of a user-friendly, agent-based simulation model for customer crowd-shipping.*
An agent-based model which incorporates all of the most notable aspects of customer crowd-shipping was designed and developed using the ANYLOGIC simulation software, PYTHON, and the PYPELINE custom library. The model was developed in a phased approach, allowing for complexity to be gradually incorporated. Importantly, the model was designed and developed by means of a bottom-up approach, which allows for the behaviour of separate entities to give rise to the overall system behaviour. To this end, each component of the model was designed to function independently — as would be the case in a real-world system. In particular, online customers are modelled as autonomous agents that may place orders, according to an assumed rate, throughout the day. In serving the online customers, the retailer's last-mile delivery system, as modelled, is capable of functioning independently, serving all online customers by means of a dedicated fleet of delivery vehicles. Within this independent function, the retailer may make decisions regarding the frequency of deliveries, as well as the solution methodology towards vehicle routing. Furthermore, in-store customers are modelled as autonomous agents, visiting the retailer according to an assumed rate. Moreover, their decision to shop at the retailer is independent of their willingness to act as an OD, which depends on certain socio-demographic characteristic of the OD and the incentive offered. The retailer has a degree of control over the customer crowd-shipping, in the form of their approach to proposing customers as crowd-shipping candidates, as well as in their choice of incentive scheme parameter values. The retailer does not, however, have the ability to control ODs or utilise them in the way that employees are utilised. Moreover, the retailer has no pre-emptive knowledge pertaining to the decisions of ODs, and make no assumptions about the future state of the system at any point in time. This allows for a robust and highly modular agent-based model, which may be adapted to suit a broad spectrum of decisions by the retailer, while considering a large number assumptions about the environment.

Contribution 3 *The refinement of the incentive scheme through the introduction of a novel maximum incentive parameter.*
A customer crowd-shipping initiative employs a compensation or incentive scheme to reward ODs for performing deliveries. As mentioned in §3.5.4, there has been a number of variations on the incentive scheme in the literature, with various authors emphasising the importance of a well-designed scheme. In this thesis, a maximum incentive parameter was introduced, producing a novel and useful incentive scheme. The use of a maximum incentive parameter restricts the size of the incentive offered. This maximum incentive parameter enables the crowdsourcer to offer incentives based on the deviation of an OD, while mitigating the risk of offering exorbitantly large incentives. Moreover, by restricting the size of the incentive, it becomes unnecessary to restrict the subset of ODs to whom a particular task is proposed. Thus, even if there are ODs that are willing to drive extremely far to delivery an order, it will not be to the detriment of the retailer.

Contribution 4 *A novel approach to capturing the behaviour of ODs as autonomous agents.*
In considering the principles of crowdsourcing, it is important to note that crowd members should not be modelled similar to the way in which dedicated employees are modelled. In the case of modelling the customer crowd-shipping initiative, the dedicated fleet of delivery vehicle may be activated to perform deliveries, whereas ODs are offered an opportunity to perform the task which they may accept according to their personal criteria. The model proposed in this thesis provides a novel modelling approach that emphasises the importance of the independent decision making of ODs. The OD population is modelled as an autonomous agent class, with individual OD agents that undertake trips to the retailer at a certain dynamic rate. The retailer may propose certain delivery tasks to ODs with the associated incentive to compensate for the additional distance that the task may require. The retailer, however, has no control over the decisions made by the OD agents in accepting or rejecting the offer. The ODs behave out of self-interest, only accepting offers that are perceived as beneficial according to their personal preferences with respect to their value of travel time. This novel modelling approach accentuates the dependency of the success of a customer crowd-shipping initiative on the perceived benefits to ODs.

Contribution 5 *A novel approach to the selection of online customers as candidates for crowd-shipping.*
In modelling ODs as independent agents, it is not possible for the retailer to pre-emptively assign them to a particular route or schedule. As such, focus is shifted towards identifying the deliveries that are less desirable to the dedicated fleet of delivery vehicles and proposing these deliveries to potentially be outsourced to ODs. This is more reflective of a real-world implementation of the customer crowd-shipping initiative, as the retailer has to consider the expenses incurred in the last-mile delivery system, but does not have the knowledge of future OD activities. The novel approach, proposed in this thesis, to selecting online customers as candidates for crowd-shipping performed by ODs depends on the cost-to-serve value of an online customer, the fixed incentive parameter, and a novel loss aversion parameter. The approach involves the dynamic identification of online customers that are expensive to serve, and the subsequent proposal of the orders associates with these online customers to be served by ODs. The loss aversion parameter governs the process of outsourcing online customers, allowing retailers to customise their outsourcing strategy. As described in §6.3.2, retailers may vary their approach from liberal to more conservative, based on their particular preference of last-mile delivery improvements (*i.e.* customer waiting time *versus* delivery savings). It was furthermore shown that the selected value of the loss aversion parameter has a significant influence on the performance of the customer crowd-shipping system. This validates that even though there is no guarantee that the outsourcing of the selected customers will lead to savings for the company, the probability of making a loss may be reduced through the use of the novel loss aversion parameter.

Contribution 6 *A trip-acceptance criteria involving the OD's perceived value of travel time.*
As mentioned in §3.5.4, it is frequently noted that an OD's acceptance of a trip is dependent on the incentive offered, although few models found in the literature incorporate this aspect. The models discussed consider criteria, such as a maximum detour distance or detour time, or assigns a random probability of a trip being accepted. The most sophisticated approach hitherto proposed considers the estimated cost incurred to the OD compared to the compensation offered, although the required incentive is known pre-emptively and may be exploited in the model solution. In the model proposed in this thesis, the acceptance criterion of ODs are modelled as an autonomous process of which the outcome is not known in advance. In particular, an OD compares the incentive offered by the retailer to

the value of the additional time required to deliver the order. This process occurs each time an OD decides to visit the retailer and is presented with a list of crowd-shipping candidates with the associated incentives from which they may select an order to deliver. The incorporation of microeconomic theory and transport economic theory provides a novel approach to this acceptance criterion.

Contribution 7 *The design and development of a GUI that may serve as a decision support system for modelling customer crowd-shipping.*
The agent-based model developed in this thesis may be utilised by means of the user-friendly GUI detailed in §4.7. The GUI may serve as a basic decision support platform which may facilitate investigations of various scenarios and their perceived real-time effects. It allows for the retailer to specify various parameters relating to their traditional last-mile delivery strategy and their customer crowd-shipping strategy. Furthermore, the model user may consider the assumptions relating to the geographical distribution of online customers, the arrival rate of orders and ODs alike, the socio-demographic factors of ODs, and the cost related to the use of delivery vehicles. Finally, it allows for real-time analysis of the model output, while tracking and displaying the relevant KPIs as the model is executed.

7.3 Suggestions for future work

The research conducted in this thesis serves as an extension of the current literature and also as a departure point for further research into the domain of customer crowd-shipping. A number of suggestions for possible future avenues of research are therefore presented in this section. These relate to various aspects of the research, and may be classified as general recommendations pertaining to customer crowd-shipping models, recommendations pertaining to vehicle routing aspects within a customer crowd-shipping model, and recommendations pertaining to modelling the behaviour of ODs.

7.3.1 General recommendations for future work in modelling customer crowd-shipping

The following recommendations are made with respect to possible future considerations, investigations, and modelling approaches toward a better understanding of customer crowd-shipping.

Proposal 1 *Investigate an incentive scheme that is based on the online customer's cost-to-serve value.*
In the proposed model, the candidates for crowd-shipping are selected by considering their cost-to-serve values, whereas each associated incentive offered is calculated based on the OD's deviation. An alternative approach involves only considering the cost-to-serve value of the particular online customer proposed as a candidate for crowd-shipping. The offered incentive is therefore independent of the OD in question and considers only the cost incurred to the retailer. This approach may be easier to implement and understand, as the same amount is offered to all ODs. The risk to such an approach, however, is that ODs are offered an amount as incentive that is independent of their deviation. As such, ODs may frequently be overpaid for a minor deviation.

Proposal 2 *Consider the consumer surplus of ODs as a KPI.*
The difference between the value of the incentive paid to an OD and their WTA for a particular offered trip (*i.e.* their perceived monetary value of the additional travel time),

may be interpreted as the consumer surplus. From the perspective of the retailer, consumer surplus is wasteful, as the amount offered is greater than the amount for which the OD is willing to perform the task. This may especially be prevalent with incentive schemes that are not bound to the OD's deviation (*e.g.* as mentioned in Proposal 1). The consumer surplus as a KPI may be conflicting with the reduction in delivery cost and waiting time. As such, finding a balance between these three KPIs may result in a customer crowd-shipping system wherein a sufficient number of ODs is willing to participate, while the retailer avoids overspending that results in consumer surplus.

Proposal 3 *Incorporate online customer involvement on the crowdsourcing platform.*
In the proposed model, it is assumed that there are no failed deliveries. In reality, however, absence of a recipient is a major problem in the context of attended last-mile deliveries, as discussed in §2.2.1. The proposal is therefore to model the involvement of online customers in the delivery process, which would be facilitated by the crowdsourcing platform in a real-world implementation. Once an OD has selected an order, the associated online customer may need to confirm their availability to receive their order and actively agree to their order being delivered by an OD. Within the agent-based model, this may be enabled through the use of conditional messages between the respective agents. As such, the probabilistic nature of online customer absence may be incorporated into the model. Moreover, the involvement of the online customers in the delivery process may reduce the risk of failed deliveries in a real-world system.

Proposal 4 *Consider the carbon emissions of the last-mile logistics system in incorporating customer crowd-shipping.*
As mentioned in §2.3.4.1, a large driver of the collaborative economy is the movement towards a more environmentally friendly and sustainable system of consumption. Customer crowd-shipping has the potential to reduce the carbon emissions of retailers by replacing trips by a dedicated delivery vehicle with the existing trips of ODs. It is therefore proposed to introduce an additional KPI, considering the carbon emissions related to both the delivery vehicles and the additional distance travelled by ODs. This KPI may be influenced by the type of delivery vehicle utilised by the retailer, as well as the efficiency of the delivery routes. On the other hand, this may also be influenced by the type of transport used by the OD. The introduction of this KPI may significantly influence the manner in which deliveries are outsourced, potentially leaning towards an increasingly liberal approach in selecting online customers as candidates for crowd-shipping. Furthermore, the retailer may adapt their crowd-shipping strategy by giving preference to ODs who utilise more sustainable forms of transport, such as public transport, electric vehicles, bicycles, or even walking.

Proposal 5 *Investigate the effect of the geographical distribution of ODs.*
In §6.3.1, the effect of the geographical distribution of online customers is investigated in isolation, while assuming that the geographical distribution of ODs is uniformly randomly distributed for all scenarios considered. In reality, however, the distribution of ODs would typically coincide with that of online customers, given that the entire range of customers of the retailer are embedded in the same geographical region. It is therefore proposed to investigate the performance of a customer crowd-shipping initiative for different geographical distribution of ODs.

Proposal 6 *Conduct a simulation optimisation.*
Given the large number of parameters that may be considered as user input, it is extremely difficult to evaluate the exhaustive set of alternative configurations. Moreover, it is challenging to determine which combination of parameter values leads to the best performance

of the customer crowd-shipping initiative under various conditions. As such, it is proposed to take an optimisation approach finding the optimal set of decision values that leads to the best performance of the system for a range of conditions.

Proposal 7 *Develop a dynamic schedule for the retailer's input parameters.*
In the current implementation, the retailer has control over a number of input parameters relating to their last-mile delivery system. These parameter values, however, are static and may only be defined at the initialisation of a simulation run. Conversely, the benefit gained from outsourcing a customer is not static and may depend on the situation at hand. For instance, the time until the next set of scheduled deliveries may influence how beneficial it is to outsource an online customer. If the dedicated set of deliveries are far into the future, the reduction in waiting time is significant if an order is outsourced, compared to the case where the set of deliveries are imminent. Similarly, the true cost-to-serve value of a customer is only known at the instantiation of the set of dedicated deliveries, whereas the estimated cost-to-serve value is less accurate the longer the time until the scheduled deliveries occur.

As such, it is proposed to devise a system where parameter values may take on a variable value that changes depending on the current scenario. For example, large incentives may be offered if the dedicated set of deliveries are far into the future, with a more liberal approach to selecting customers as crowd-shipping candidates. As the set of scheduled deliveries approaches, the offered incentive may be reduced and more strict criteria used in selecting crowd-shipping candidates. Finally, given the vast number of parameters and influential factors, it is proposed to consider reinforcement learning in dynamically determining the optimal set of parameter values. This will allow for the system to learn, during model execution, which set of parameters results in the best performance of the customer crowd-shipping model, given the current state of the system.

7.3.2 Recommendations for future work pertaining to vehicle routing within a customer crowd-shipping model

A number of recommendations are made with respect to the vehicle routing aspects considered in the customer crowd-shipping model. This pertains to the formulation of the VRP, the assumptions regarding the constituent elements, as well as the solution methodologies utilised to solve the VRP during model execution. Finally, recommendations pertaining to the subsequent calculation of the cost-to-serve values are additionally proposed for future work.

Proposal 8 *Incorporate time-based considerations for the last-mile deliveries.*
The current model does not consider time windows, implying that customers may be served at any time. Time windows can be incorporated into the proposed agent-based model in various ways. First, the introduction of a maximum delivery lead time may be incorporated. This would be ideal for companies which last-mile delivery system includes same-day deliveries with time-critical orders. Alternatively, each order placed may have an associated time window wherein it should be delivered. The CVRP utilised throughout the model may then be extended to a VRPTW to ensure adherence to the time constraints imposed. Moreover, potential crowd-shipping candidates may be added to the regular delivery route once the risk of a late delivery becomes too large.

Proposal 9 *Add complexity with respect to the delivery vehicles.*
In the proposed model a set of homogeneous delivery vehicles with a fixed distance-based

cost is assumed. The model may be extended to incorporate a number of additional complexities with respect to the delivery vehicles. First, the cost of utilising a delivery vehicle may be partitioned into the fixed cost of a vehicle, a cost per unit distance travelled, as well as the wage rate of the driver. Such a cost structure may provide new insights into the manner in which crowd-shipping should accompany regular deliveries. Furthermore, a set of heterogeneous delivery vehicles may be considered, with variations in the capacity and speed, while additionally accounting for the cost structure of the different vehicles. The agent-based approach lends itself towards such categorisation of delivery vehicles. Finally, the assumed infinite availability of delivery vehicles may be challenged. The scarcity of delivery vehicles may further complement the real-world nature of the model.

Proposal 10 *Utilise more sophisticated vehicle routing solution methodologies.*
In the current implementation of the customer crowd-shipping model, the CVRP may be solved by means of the saving and sweep algorithms. These heuristics provide solutions within a short timeframe, although the resulting routes are often sub-optimal. It is therefore proposed to investigate the implementation of more sophisticated CVRP solution methodologies in the customer crowd-shipping model. Solutions of a higher quality would not only result in a lower delivery vehicle costs, but also in a more robust and accurate estimate of the cost-to-serve values. Finally, the robustness of the benefit in implementing customer crowd-shipping may be evaluated when implemented in conjunction with a highly sophisticated solution methodology to the CVRP.

Proposal 11 *Consider the future demand when calculating the cost-to-serve values.*
The current process for calculating cost-to-serve values is performed each time an order is placed, with no regard for the orders that are likely to realise in the future. The true cost to serve a particular customer, however, is only known at the instantiation of the subsequent set of dedicated deliveries. This presents the problem that online customers may be outsourced based on their current cost-to-serve value, which may not be a true representation of the actual cost required. It is therefore proposed to devise a methodology in estimating the cost to serve a particular customer by predicting the orders that are likely to be placed in the future. By using the estimated value, rather than the cost-to-serve value for that specific moment in time, a more sophisticated approach to candidate selection may be devised.

Proposal 12 *Consider the savings resulting from the outsourcing of a combination of online customers.*
The current methodology for calculating the cost-to-serve value of a customer considers the cost of the delivery route with and without the particular customer. When considering a combination of customers, however, the calculated cost-to-serve values may give a false impression of the savings that may be realised by outsourcing these customers. Consider the case where two online customers are located relatively close to one another, but far from the depot. The current methodology would suggest that both customers are relatively inexpensive to serve. If both customers are outsourced, however, a larger reduction in delivery cost may be realised, contrary to what would be deduced from their individual cost-to-serve values. As such, it is proposed to calculate the potential savings of outsourcing a combination of customers in addition to the individual cost-to-serve values. This may be incorporated particularly well for a system where ODs are able to deliver multiple orders.

7.3.3 Recommendations for future work pertaining to the modelling of OD behaviour

The final category of recommendations pertains to the behaviour of ODs. A novel approach is proposed in the agent-based model, with the decision making of independent OD agents relying on their willingness to trade time for money. This initial approach, involving the incorporation of microeconomic theory and transport economics, leaves ample room for improvement and may serve as a departure point for various avenues of research.

Proposal 13 *Allow for ODs to deliver multiple orders.*
The proposed model allows for an OD to deliver only one order per trip from the depot. In reality, an OD may be able and willing to deliver multiple orders. The proposal is therefore to extend the model to allow for multiple orders to be delivered by any single OD. This approach may be complemented by considering the cost of serving a combination of customers, as suggested in Proposal 12.

Proposal 14 *Improve the value of time functions for ODs.*
A number of simplifying assumptions are made to model an OD's willingness to trade travel time for money. One such assumption is that the indifference curve of travel time and money may be modelled as perfect substitutes. This implies that an OD's willingness to trade is independent of the amount of travel time or money traded. In reality, as well as in the more general form of indifference curves, consumers have preference for a more diverse bundle of goods. As such, although the assumption of perfect substitutes provides a good departure point in modelling an OD's value of time, it is recommended to consider the value of time functions according to microeconomic principles in reflecting the reality more adequately.

Proposal 15 *Incorporate the distance of a trip as a factor that influences the value of time.*
As mentioned in §3.4.3, the distance of a trip is found to significantly influence the value that individuals associate to their travel time. The model proposed in this thesis primarily considers the effect of income and congestion. It is proposed to incorporate the additional effect of the distance of a trip on the value that ODs may place on their travel time. This may refine the modelling of OD behaviour and therefore contribute to a more realistic model.

References

[1] Abrantes PA & Wardman MR, 2011, *Meta-analysis of UK values of travel time: An update*, Transportation Research Part A: Policy and Practice, **45(1)**, pp. 1–17.

[2] Acquier A, Carbone V & Massé D, 2019, *How to create value(s) in the sharing economy: Business models, scalability, and sustainability*, Technology Innovation Management Review, **9(2)**, pp. 5–25.

[3] Adepetu A, Ahmed K, Abd Y, Zaabi A & Svetinovic D, 2012, *CrowdREquire: A requirements engineering crowdsourcing platform*, Proceedings of the AAAI Spring Symposium: Wisdom of the crowd, Palo Alto (CA), pp. 2–7.

[4] Airbnb, 2021, *How Airbnb works*, [Online], [Cited September 2021], Available from `https://www.airbnb.com/help/topic/1378/how-airbnb-works`.

[5] Ali F, 2021, *US ecommerce grows* 32.4% *in 2020*, [Online], [Cited March 2021], Available from `https://www.digitalcommerce360.com/article/us-ecommerce-sales`.

[6] Allen J, Piecyk M, Piotrowska M, McLeod F, Cherrett T, Ghali K, Nguyen T, Bektas T, Bates O, Friday A, Wise S & Austwick M, 2018, *Understanding the impact of e-commerce on last-mile light goods vehicle activity in urban areas: The case of London*, Transportation Research Part D: Transport and Environment, **61**, pp. 325–338.

[7] Allio RJ, 2004, *CEO interview: The InnoCentive model of open innovation*, Strategy & Leadership, **32(4)**, pp. 4–9.

[8] Amazon Mechanical Turk, 2021, *Overview*, [Online], [Cited September 2021], Available from `https://www.mturk.com/`.

[9] Archetti C, Savelsbergh M & Speranza MG, 2016, *The vehicle routing problem with occasional drivers*, European Journal of Operational Research, **254(2)**, pp. 472–480.

[10] Armstrong P, Garrido R & Ortúzar JDD, 2001, *Confidence intervals to bound the value of time*, Transportation Research Part E: Logistics and Transportation Review, **37(2-3)**, pp. 143–161.

[11] Arora JS, 2017, *Introduction to optimum design*, 4th Edition, Academic Press, Boston (MA).

[12] Arslan AM, Agatz N, Kroon L & Zuidwijk R, 2018, *Crowdsourced delivery — A dynamic pickup and delivery problem with ad-hoc drivers*, Transportation Science, **53(1)**, pp. 222–235.

[13] Baba Y & Kashima H, 2013, *Statistical quality estimation for general crowdsourcing tasks*, Proceedings of the 19^{th} ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago (IL), pp. 554–562.

[14] Baindur D & Macário RM, 2013, *Mumbai lunch box delivery system: A transferable benchmark in urban logistics?*, Research in Transportation Economics, **38(1)**, pp. 110–121.

[15] Baker BM, 1992, *Further improvements to vehicle routeing heuristics*, Journal of the Operational Research Society, **43(10)**, pp. 1009–1012.

[16] Baker BM & Sheasby J, 1999, *Extensions to the generalised assignment heuristic for vehicle routing*, European Journal of Operational Research, **119(1)**, pp. 147–157.

[17] Baldacci R, Mingozzi A & Roberti R, 2012, *Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints*, European Journal of Operational Research, **218(1)**, pp. 1–6.

[18] Balinski ML & Quandt RE, 1964, *On an integer program for a delivery problem*, Operations Research, **12(2)**, pp. 300–304.

[19] Banerjee A & Siemens F, 2015, *Logistics of E-Groceries.de*, Proceedings of the Innovations and Strategies for Logistics and Supply Chains: Technologies, Business Models and Risk Management, Berlin, pp. 91–116.

[20] Banks J, 1998, *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, Wiley-Interscience, New York (NY).

[21] Barr A & Wohl J, 2013, *Exclusive: Wal-Mart may get customers to deliver packages to online buyers*, [Online], [Cited March 2020], Available from `https://www.reuters.com/article/us-retail-walmart-delivery/exclusive-wal-mart-may-get-customers-to-deliver-packages-to-online-buyers-idUSBRE92R03820130328`.

[22] Bederson B & Quinn A, 2011, *Web workers unite! Addressing challenges of online labourers*, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vancouver (BC), pp. 97–106.

[23] Bensinger G, 2015, *Amazon's next delivery drone: You*, [Online], [Cited March 2020], Available from `https://www.wsj.com/articles/amazon-seeks-help-with-deliveries-1434466857`.

[24] Bingi P, Mir A & Khamalah J, 2000, *The challenges facing global e-commerce*, Information Systems Management, **17(4)**, pp. 22–30.

[25] Bjerkan KY, Bjørgen A & Hjelkrem OA, 2020, *E-commerce and prevalence of last mile practices*, Transportation Research Procedia, **46**, pp. 293–300.

[26] Blum C, 2005, *Ant colony optimization: Introduction and recent trends*, Physics of Life Reviews, **2(4)**, pp. 353–373.

[27] Bonabeau E, 2002, *Agent-based modeling: Methods and techniques for simulating human systems*, Proceedings of the 3^{rd} Colloquium of the National Academy of Science, Irvine (CA), pp. 7280–7287.

[28] Borshchev A, 2013, *The big book of simulation modeling: Multimethod modeling with AnyLogic 6*, AnyLogic North America, Lisle (IL).

[29] Borshchev A & Filippov A, 2004, *From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools*, Proceedings of the 22^{nd} International Conference of the System Dynamics Society, Oxford, pp. 25–29.

[30] Bosona T, 2020, *Urban freight last mile logistics — Challenges and opportunities to improve sustainability: A literature review*, Sustainability, **12(21)**, pp. 1–20.

[31] Botsman R & Rogers R, 2011, *What's mine is yours: How collaborative consumption is changing the way we live*, Collins, London.

[32] Boyer KK, Prud'homme AM & Chung W, 2009, *The last mile challenge: Evaluating the effects of customer density and delivery window patterns*, Journal of Business Logistics, **30(1)**, pp. 185–201.

[33] Brabham DC, 2008, *Crowdsourcing as a model for problem solving: An introduction and cases*, Convergence: The International Journal of Research into New Media Technologies, **14(1)**, pp. 75–90.

[34] Brabham DC, 2008, *Moving the crowd at iStockphoto: The composition of the crowd and motivations for participation in a crowdsourcing application*, First Monday **13(6)**, [1396-0466], [Cited October, 2020], Available from `https://firstmonday.org/ojs/index.php/fm/article/view/2159/1969`.

[35] Bramel J & Simchi-Levi D, 1995, *A location based heuristic for general routing problems*, Operations Research, **43(4)**, pp. 649–660.

[36] Bullnheimer B, Hartl RF & Strauss C, 1999, *Applying the ant systems to the vehicle routing problem*, pp. 285–296 in Voss S, Martello S, Osman I & Roucairol C (Eds), *Meta-heuristics: Advances and trends in local search paradigms for optimization*, Springer, Boston (MA).

[37] Byl R, 2016, *Bpost launches bringr: Uber-inspired parcel delivery app*, [Online], [Cited February 2021], Available from `https://trends.knack.be/economie/bedrijven/bpost-lanceert-bringr-op-uber-geinspireerde-app-om-pakjes-te-bezorgen/article-normal-717843.html?cookie_check=1614085303`.

[38] Carbone V, Rouquet A & Roussat C, 2017, *The rise of crowd lgistics: A new way to co-create logistics value*, Journal of Business Logistics, **38(4)**, pp. 238–252.

[39] Cesario FJ, 1976, *Value of time in recreation benefit studies*, Land Economics, **52(1)**, pp. 32–41.

[40] Chanal V & Caron-Fasan M, 2008, *How to invent a new business model based on crowdsourcing: The crowdspirit ® case*, Proceedings of the 17^{th} Conférence Internationale de Management Stratégique, Nice, pp. 1–27.

[41] Chen C, Cheng S.-F, Gunawan A, Misra A, Dasgupta K & Chander D, 2014, *TRACCS: Trajectory-aware coordinated urban crowd-sourcing*, Proceedings of the 2^{nd} AAAI Conference on Human Computation and Crowdsourcing HCOMP, Pittsburgh (KS), pp. 30–40.

[42] Chen MK, 2016, *Dynamic pricing in a labor market: Surge pricing and flexible work on the Uber platform*, Proceedings of the 2016 ACM Conference on Economics and Computation, Maastricht, pp. 455–456.

[43] Chen P & Chankov SM, 2017, *Crowdsourced delivery for last-mile distribution: An agent-based modelling and simulation approach*, Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management, Singapore, pp. 1271–1275.

[44] Christofides N & Eilon S, 1969, *An algorithm for the vehicle-dispatching problem*, Journal of the Operational Research Society, **20(3)**, pp. 309–318.

[45] Chung KL, 1974, *A Course in Probability Theory*, 2^{nd} Edition, Academic Press, New York (NY).

[46] Clarke G & Wright JW, 1964, *Scheduling of vehicles from a central depot to a number of delivery points*, Operations Research, **12(4)**, pp. 568–581.

[47] Cohen B & Muñoz P, 2016, *Sharing cities and sustainable consumption and production: Towards an integrated framework*, Journal of Cleaner Production, **134**, pp. 87–97.

[48] Colorni A, Dorigo M & Maniezzo V, 1991, *Distributed optimization by ant colonies*, Proceedings of the 1^{st} European Conference on Artificial Life, Paris, pp. 134–142.

[49] Comert S, Yazgan H, Kir S & Yener F, 2018, *A cluster first-route second approach for a capacitated vehicle routing problem: A case study*, International Journal of Procurement Management, **11(4)**, pp. 399–419.

[50] Coppola D, 2019, *Primary reason for digital shoppers in the United States to abandon their carts as of November 2018*, [Online], [Cited April 2020], Available from `https://www.statista.com/statistics/379508/primary-reason-for-digital-shoppers-to-abandon-carts/`.

[51] Coppola D, 2021, *E-commerce share of total global retail sales from 2015 to 2024*, [Online], [Cited November 2021], Available from `https://www.statista.com/statistics/534123/e-commerce-share-of-retail-sales-worldwide/`.

[52] Coppola D, 2021, *E-commerce worldwide – Statistics & facts*, [Online], [Cited November 2021], Available from `https://www.statista.com/topics/871/online-shopping/`.

[53] Cordeau JF, Gendreau M, Hertz A, Laporte G & Sormany JS, 2005, *New heuristics for the vehicle routing problem*, pp. 279–297 in Langevin A & Riopel D (Eds), *Logistics Systems: Design and Optimization*, Springer, Boston (MA).

[54] Cordeau J, Gendreau M, Laporte G, Potvin J & Semet F, 2002, *A guide to vehicle routing heuristics*, Journal of the Operational Research Society, **53(5)**, pp. 512–522.

[55] Couchsurfing, 2021, *About us*, [Online], [Cited September 2021], Available from `https://about.couchsurfing.com/about/about-us/`.

[56] Craigs-list, 2021, *About*, [Online], [Cited September 2021], Available from `https://www.craigslist.org/about/?lang=en&cc=gb`.

[57] Cramer-Flood E, 2020, *Global e-commerce 2020*, [Online], [Cited June 2020], Available from `https://www.emarketer.com/content/global-ecommerce-2020`.

[58] Crescenzi V, Merialdo P & Qiu D, 2013, *Wrapper generation supervised by a noisy crowd*, Proceedings of the 1^{st} VLDB Workshop on Databases and Crowdsourcing, Trento, pp. 8–13.

[59] Crook JH, 1970, *Sources of cooperation in animals and man*, Social Science Information, **9(1)**, pp. 27–48.

[60] CrunchBase, 2021, *Zilok*, [Online], [Cited September 2021], Available from `https://www.crunchbase.com/organization/zilok`.

[61] CSP Online, 2016, *From storefronts to search engines: A history of e-commerce*, [Online], [Cited November 2020], Available from `https://online.csp.edu/blog/business/history-of-ecommerce/`.

[62] Dadhich P, 2021, Research Associate at *Centre for Sustainable Road Freight, Heriot-Watt University*, [Personal Communication], Contactable at `p.dadhich@hw.ac.uk`.

[63] Dahle L, Andersson H & Marielle C, 2017, *The vehicle routing problem with dynamic occasional drivers*, Proceedings of the 8^{th} International Conference on Computational Logistics, Southampton, pp. 49–63.

[64] Dahle L, Andersson H, Christiansen M & Speranza MG, 2019, *The pickup and delivery problem with time windows and occasional drivers*, Computers and Operations Research, **109**, pp. 122–133.

[65] Dantzig GB & Ramser JH, 1959, *The truck dispatching problem*, Management Science, **6(1)**, pp. 80–91.

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

[67] De Borger B & Fosgerau M, 2008, *The trade-off between money and travel time: A test of the theory of reference-dependent preferences*, Journal of Urban Economics, **64(1)**, pp. 101–115.

[68] Dhillon S & Wu K, 2021, *Delivery 2.0: How on-demand meal service will become something far bigger*, [Online], [Cited February 2021], Available from `https://www.fastcompany.com/90604082/future-of-on-demand-meal-delivery-ghost-kitchens-postmates-doordash-uber-eats`.

[69] Difallah DE, Demartini G & Cudré-Mauroux P, 2012, *Mechanical cheat: Spamming schemes and adversarial techniques on crowdsourcing platforms*, Proceedings of the 1^{st} International Workshop on Crowdsourcing Web Search, CrowdSearch, Lyon, pp. 20–25.

[70] Djelassi S & Decoopman I, 2013, *Customers' participation in product development through crowdsourcing: Issues and implications*, Industrial Marketing Management, **42(5)**, pp. 683–692.

[71] Dorigo M, Maniezzo V & Colorni A, 1996, *Ant system: Optimization by a colony of cooperating agents*, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, **26(1)**, pp. 29–41.

[72] Dorman P, 2014, *Microeconomics*, Springer, Heidelberg.

[73] dos Santos VF, Sabino LR, Morais GM & Goncalves CA, 2017, *E-Commerce: A short history follow-up on possible trends*, International Journal of Business Administration, **8(7)**, pp. 130–138.

[74] eBay, 2021, *eBay: Company overview*, [Online], [Cited September 2021], Available from `https://www.ebayinc.com/company/`.

[75] eMarketer, 2013, *Ecommerce sales topped $1 trillion for first time in 2012*, [Online], [Cited November 2020], Available from `https://www.emarketer.com/Article/Ecommerce-Sales-Topped-1-Trillion-First-Time-2012/1009649`.

[76] Erento, 2021, *About us*, [Online], [Cited September 2021], Available from `https://www.erento.com/info/ueber-erento/`.

[77] Esper T, Jensen T, Turnipseed F & Burton S, 2003, *The last mile: An examination of effects of online retail delivery strategies on consumers*, Journal of Business Logistics, **24(2)**, pp. 177–203.

[78] Estellés-Arolas E & González-Ladrón-de-Guevara F, 2012, *Towards an integrated crowdsourcing definition*, Journal of Information Science, **38(2)**, pp. 198–200.

[79] European Environment Agency, 2016, *Occupancy rates of passenger vehicles*, [Online], [Cited February 2021], Available from `https://www.eea.europa.eu/data-and-maps/indicators/occupancy-rates-of-passenger-vehicles/occupancy-rates-of-passenger-vehicles`.

[80] Feinberg RA, 1986, *Credit cards as spending facilitating stimuli: A conditioning interpretation*, Journal of Consumer Research, **13(3)**, pp. 348–356.

[81] Felch V, Karl D, Asdecker B, Niedermaier A & Sucky E, 2019, *Reconfiguration of the last mile: Consumer acceptance of alternative delivery concepts*, pp. 157–171 in Bierwirth C, Kirschstein T & Sackmann D (Eds), *Logistics Management*, Springer International Publishing, Cham.

[82] Fisher ML & Jaikumar R, 1981, *A generalized assignment heuristic for vehicle routing*, Networks, **11(2)**, pp. 109–124.

[83] Forrester JW, 1993, *System dynamics and the lessons of 35 years*, pp. 199–240 in De Greene KB (Ed), *A systems-based aproach to policymaking*, Springer, Boston (MA).

[84] Fraternali P, Castelletti A, Soncini-Sessa R, Vaca C & Rizzoli AE, 2012, *Putting humans in the loop: Social computing for water resources management*, Environmental Modelling & Software, **37**, pp. 68–77.

[85] Fry R, 2013, *Young adults after the recession: Fewer homes, fewer cars, less debt*, (Unpublished) Technical Report, Pew Research Center, Washington (DC).

[86] Gao H, Barbier G & Goolsby R, 2011, *Harnessing the crowdsourcing power of social media for disaster relief*, Intelligent Systems, **26(3)**, pp. 10–14.

[87] Gaskell TJ, 1967, *Bases for vehicle fleet scheduling*, Journal of the Operational Research Society, **18(3)**, pp. 281–295.

[88] Gatta V, Marcucci E, Le Pira M, Inturri G, Ignaccolo M & Pluchino A, 2020, *E-groceries and urban freight: Investigating purchasing habits, peer influence and behaviour change via a discrete choice/agent-based modelling approach*, Transportation Research Procedia, **46**, pp. 133–140.

[89] Gaudry MJ, Jara-Diaz SR & Ortuzar JD, 1989, *Value of time sensitivity to model specification*, Transportation Research Part B, **23(2)**, pp. 151–158.

[90] Gdowska K, Viana A & Pedroso JP, 2018, *Stochastic last-mile delivery with crowdshipping*, Transportation Research Procedia, **30**, pp. 90–100.

[91] Gendreau M, Laporte G & Potvin JY, 2002, *Metaheuristics for the capacitated VRP*, pp. 129–154 in Toth P & Vigo D (Eds), *The Vehicle Routing Problem*, Society for Industrial and Applied Mathematics, Philadelphia (PA).

[92] Gevaers R, Van de Voorde E & Vanelslander T, 2009, *Characteristics of innovations in last mile logistics — Using best practices, case studies and making the link with green and sustainable logistics*, Proceedings of the European Transport Conference, Noordwijkerhout, pp. 1–21.

[93] Gevaers R, Van de Voorde E & Vanelslander T, 2011, *Characteristics and typology of last mile logistics from an innovation perspective in an urban context*, pp. 56–71 in Macharis C & Melo S (Eds), *City distribution and urban freight transport*, Edward Elgar Publishing, Cheltenham.

[94] Gevaers R, Van de Voorde E & Vanelslander T, 2014, *Cost modelling and simulation of last-mile characteristics in an innovative B2C supply chain environment with implications on urban areas and cities*, Procedia — Social and Behavioral Sciences, **125**, pp. 398–411.

[95] Ghezzi A, Gabelloni D, Martini A & Natalicchio A, 2018, *Crowdsourcing: A review and suggestions for future research*, International Journal of Management Reviews, **20(2)**, pp. 343–363.

[96] Gillett BE & Miller LR, 1974, *A heuristic algorithm for the vehicle-dispatch problem*, Operations Research, **22(2)**, pp. 340–349.

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

[98] Golden BL, Magnanti TL & Nguyen HQ, 1977, *Implementing vehicle routing algorithms*, Networks, **7**, pp. 113–148.

[99] Goncalves J, Ferreira D, Hosio S, Liu Y, Rogstadius J, Kukka H & Kostakos V, 2013, *Crowdsourcing on the spot: Altruistic use of public displays, feasibility, performance, and behaviours*, Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Zurich, pp. 753–762.

[100] González RM, 1997, *The value of time: A theoretical review*, Transport Reviews, **17(3)**, pp. 245–266.

[101] Groover MP, 2019, *Automation*, [Online], [Cited May 2020], Available from `https://www.britannica.com/technology/automation`.

[102] Gumtree, 2021, *Gumtree: About us*, [Online], [Cited September 2021], Available from `https://help.gumtree.co.za/articles/KB_Article/About-Us?vgroup1=PublicKB&vcategory1=PublicKB:Gumtree_Basics&vcategory2=About_us`.

[103] Hamari J & Ukkonen A, 2016, *The sharing economy: Why people participate in collaborative consumption*, Journal of the Association for Information Science and Technology, **67(9)**, pp. 2047–2059.

[104] Hansen D, Schone P, Corey D, Reid M & Gehring J, 2013, *Quality control mechanisms for crowdsourcing*, Proceedings of the Conference on Computer Supported Cooperative Work, San Antonio (TX), pp. 649–660.

[105] Hayes A, 2019, *Dotcom bubble*, [Online], [Cited March 2021], Available from `https://www.investopedia.com/terms/d/dotcom-bubble.asp`.

[106] Heipke C, 2010, *Crowdsourcing geospatial data*, ISPRS Journal of Photogrammetry and Remote Sensing, **65(6)**, pp. 550–557.

[107] Helsgaun K, 2000, *Effective implementation of the Lin-Kernighan traveling salesman heuristic*, European Journal of Operational Research, **126(1)**, pp. 106–130.

[108] Hirth M, Hossfeld T & Tran-Gia P, 2011, *Anatomy of a crowdsourcing platform — Using the example of microworkers.com*, Proceedings of the 5^{th} International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Asan, pp. 322–329.

[109] Hischier R, 2018, *Car vs. packaging — A first, simple (environmental) sustainability assessment of our changing shopping behaviour*, Sustainability, **10(9)**, pp. 3061.

[110] Holland JH, 1992, *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*, MIT Press, Cambridge (MA).

[111] Hosseini M, Phalp K, Taylor J & Ali R, 2014, *The four pillars of crowdsourcing: A reference model*, Proceedings of the 8^{th} International Conference on Research Challenges in Information Science, Marrakech, pp. 1–12.

[112] Howe J, 2006, *The rise of crowdsourcing*, [Online], [Cited September 2020], Available from `https://www.wired.com/2006/06/crowds/`.

[113] Innocentive, 2021, *Customer stories: NASA Kevlar*, [Online], [Cited September 2021], Available from `https://www.innocentive.com/resources/customer-stories/nasa-kevlar/`.

[114] International Transport Forum, 2019, *What is the value of saving travel time?*, (Unpublished) Technical Report 176, International Transport Forum, Paris.

[115] Janssen M, 2005, *Agent-based modeling*, pp. 155–172 in Proops J & Safonov P (Eds), *Modeling in ecological economics*, Edward Elgar Publishers, Cheltenham.

[116] Jungnickel D, 2008, *Graphs, networks and algorithms*, 3rd Edition, Springer, Heidelberg.

[117] Kahneman D, Knetsch JL & Thaler RH, 1990, *Experimental tests of the endowment effect and the coase theorem*, Journal of Political Economy, **98(6)**, pp. 1325–1348.

[118] Kämäräinen V, Saranen J & Holmström J, 2001, *The reception box impact on home delivery efficiency in the e-grocery business*, International Journal of Physical Distribution & Logistics Management, **31(6)**, pp. 414–426.

[119] Kirkpatrick S, Gelatt CD & Vecchi MP, 1983, *Optimization by simulated annealing*, Science, **220(4598)**, pp. 671–680.

[120] Klein DS, 2002, *Visionary in obscurity: Charles Stack*, [Online], [Cited November 2020], Available from `http://www.sbnonline.com/article/visionary-in-obscurity-charles-stack-operates-in-two-business-communities-151-cleveland-and-the-internet-151-and-isn-146-t-well-known-in-either-this-time-around-that-146-s-going-to-change-he-hopes/`.

[121] Kohler T, 2015, *Crowdsourcing-based business models: How to create and capture value*, California Management Review, **57(4)**, pp. 63–85.

[122] Kolmar M, 2017, *Principles of microeconomics*, Springer, Cham.

[123] Korayem L, Khorsid M & Kassem SS, 2015, *Using Grey Wolf algorithm to solve the capacitated vehicle routing problem*, IOP Conference Series: Materials Science and Engineering, **83**, pp. 1–10.

[124] Laporte G, Gendreau M, Potvin JY & Semet F, 2000, *Classical and modern heuristics for the vehicle routing problem*, International Transactions in Operational Research, **7(4-5)**, pp. 285–300.

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

[126] Law AM & Kelton WD, 2015, *Simulation modeling and analysis*, 5th Edition, McGraw-Hill, New York (NY).

[127] Lebraty JF & Lobre-Lebraty K, 2013, *Crowdsourcing: One step beyond*, Wiley, London.

[128] Lin S & Kernighan BW, 1973, *An effective heuristic algorithm for the traveling salesman problem*, Operations Research, **21(2)**, pp. 498–516.

[129] Lin S, 1965, *Computer solutions of the traveling salesman problem*, Bell System Technical Journal, **44(10)**, pp. 2245–2269.

[130] LogisticsMatter, 2013, *DHL crowd sources deliveries in Stockholm with MyWays*, [Online], [Cited September 2020], Available from `https://logisticsmatter.com/dhl-crowd-sources-deliveries-in-stockholm-with-myways/`.

[131] Maat K & Konings R, 2018, *Accessibility or innovation? Store shopping trips versus online shopping*, Transportation Research Record, **2672(50)**, pp. 1–10.

[132] Macal C & North M, 2014, *Introductory tutorial: Agent-based modeling and simulation*, Proceedings of the 2014 Winter Simulation Conference, Argonne (IL), pp. 1539–1548.

[133] Maclean D, Yoshida K, Edwards A, Crossman L, Clavijo B, Clark M, Swarbreck D, Bashton M, Chapman P, Gijzen M, Caccamo M, Downie A, Kamoun S & Saunders D, 2013, *Crowdsourcing genomic analyses of ash and ash dieback — Power to the people*, GigaScience, **2(1)**, pp. 2.

[134] Maria A, 1997, *Introduction to modeling and simulation*, Proceedings of the 29th Winter Simulation Conference, Atlanta (GA), pp. 7–13.

[135] Markoff J, 2006, *What the dormouse said: How the sixties counterculture shaped the personal computer industry*, Penguin Books, New York (NY).

[136] Martin CJ, 2016, *The sharing economy: A pathway to sustainability or a nightmarish form of neoliberal capitalism?*, Ecological Economics, **121**, pp. 149–159.

[137] Mattioli D & Herrera S, 2020, *Amazon's sales jump as coronavirus prompts surge in online shopping*, [Online], [Cited March 2020], Available from `https://www.wsj.com/articles/amazons-sales-jump-as-coronavirus-prompts-surge-in-online-shopping-11588278740?cx_testId=3&cx_testVariant=cx_2&cx_artPos=4#cxrecs_s`.

[138] Mehmann J, Frehe V & Teuteberg F, 2015, *Crowd logistics — A literature review and maturity model*, Proceedings of the 20th International Conference of Logistics, Hamburg, pp. 117–145.

[139] Melacini M, Perotti S, Rasini M & Tappia E, 2018, *E-fulfilment and distribution in omni-channel retailing: A systematic literature review*, International Journal of Physical Distribution and Logistics Management, **48(4)**, pp. 391–414.

[140] Mendoza JE, Guéret C, Hoskins M, Lobit H, Pillac V, Vidal T & Vigo D, 2014, *VRP-REP: The vehicle routing community repository*, [Online], [Cited October 2021], Available from `http://www.vrp-rep.org/datasets.html`.

[141] Mill JS, 1874, *On the definition and method of political economy*, Essay 5 in Mill JS (Ed), *Essays on some unsettled questions of political economy*, Longmans, Green, Reader, and Dyer, London.

[142] Mitchell JE, 2002, *Branch-and-cut algorithms for combinatorial optimization problems*, pp. 65–77 in Pitsoulis LS & Resende MG (Eds), *Handbook of applied optimization*, Oxford University Press, Oxford.

[143] Mittal T, 2018, *How the Alibaba Group grew from a small apartment to a global e-commerce giant*, [Online], [Cited November 2020], Available from `https://yourstory.com/2018/02/the-story-behind-the-alibaba-group`.

[144] Mladenow A, Bauer C & Strauss C, 2016, *"Crowd logistics": The contribution of social crowds in logistics activities*, International Journal of Web Information Systems, **12(3)**, pp. 379–396.

[145] Mladenow A, Bauer C & Strauss C, 2015, *Crowdsourcing in logistics: Concepts and applications using the social crowd*, Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services, Brussels, pp. 244–251.

[146] Mortensen J, Musen M & Noy N, 2013, *Crowdsourcing the verification of relationships in biomedical ontologies*, Proceedings of the AMIA Annual Symposium, Online, pp. 1020–1029.

[147] Muñoz-Villamizar A, Montoya-Torres JR & Vega-Mejía CA, 2015, *Non-collaborative versus collaborative last-mile delivery in urban systems with stochastic demands*, Proceedings of the 7th Industrial Product-Service Systems Conference, Saint-Etienne, pp. 263–268.

[148] Nassauer S, 2016, *Wal-Mart to test grocery delivery with Uber and Lyft*, [Online], [Cited February 2021], Available from `https://www.wsj.com/articles/wal-mart-to-test-grocery-delivery-with-uber-and-lyft-1464926454`.

[149] Nisar TM & Prabhakar G, 2017, *What factors determine e-satisfaction and consumer spending in e-commerce retailing?*, Journal of Retailing and Consumer Services, **39**, pp. 135–144.

[150] O'Sullivan D & Haklay M, 2000, *Agent-based models and individualism: Is the world agent-based?*, Environment and Planning, **32(8)**, pp. 1409–1425.

[151] Olsson J, Hellström D & Pålsson H, 2019, *Framework of last mile logistics research: A systematic review of the literature*, Sustainability, **11(24)**, pp. 1–25.

[152] Öncan T, 2007, *A survey of the generalized assignment problem and its applications*, Information Systems and Operational Research, **45(3)**, pp. 123–141.

[153] Osman IH, 1993, *Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem*, Annals of Operations Research, **41**, pp. 421–451.

[154] Owyang J, 2013, *The collaborative economy*, (Unpublished) Technical Report, Altimeter Group, San Mateo (CA).

[155] Pan S, Giannikas V, Han Y, Grover-Silva E & Qiao B, 2017, *Using customer-related data to enhance e-grocery home delivery*, Industrial Management and Data Systems, **117(9)**, pp. 1917–1933.

[156] Petropoulos G, 2017, *An economic review of the collaborative economy*, Bruegel Policy Contribution, **5**, pp. 1–17.

[157] Prelec D & Simester D, 2001, *Always leave home without it: A further investigation of the credit-card effect on willingness to pay*, Marketing Letters, **12(1)**, pp. 5–12.

[158] Price JA, 1975, *Sharing: The integration of intimate economies*, Anthropologica, **17(1)**, pp. 3–27.

[159] Prins C, 2004, *A simple and effective evolutionary algorithm for the vehicle routing problem*, Computers and Operations Research, **31(12)**, pp. 1985–2002.

[160] Punel A & Stathopoulos A, 2017, *Modeling the acceptability of crowdsourced goods deliveries: Role of context and experience effects*, Transportation Research Part E: Logistics and Transportation Review, **105**, pp. 18–38.

[161] Pureza VM & França PM, 1991, *Vehicle routing problems via tabu search metaheuristic*, (Unpublished) Technical Report, Centre for Research on Transportation, Montreal.

[162] Rai HB, Verlinde S, Merckx J & Macharis C, 2017, *Crowd logistics: An opportunity for more sustainable urban freight transport?*, European Transport Research Review, **9(3)**, pp. 1–13.

[163] Ranard B, Ha Y, Meisel Z, Asch D, Hill S, Becker L, Seymour A & Merchant R, 2013, *Crowdsourcing — Harnessing the masses to advance health and medicine, a systematic review*, Journal of General Internal Medicine, **29(1)**, pp. 187–203.

[164] Ranieri L, Digiesi S, Silvestri B & Roccotelli M, 2018, *A review of last mile logistics innovations in an externalities cost reduction vision*, Sustainability, **10(3)**, pp. 1–18.

[165] Rautela A, Sharma SK & Bhardwaj P, 2019, *Distribution planning using capacitated clustering and vehicle routing problem: A case of Indian cooperative dairy*, Journal of Advances in Management Research, **16(5)**, pp. 781–795.

[166] Reimann M, Doerner K & F HR, 2004, *D-ants: Savings based ants divide and conquer the vehicle routing problem*, Computers and Operations Research, **31(4)**, pp. 563–591.

[167] Reinhardt A, 2001, *Tesco bets small and wins big*, [Online], [Cited November 2020], Available from `https://www.bloomberg.com/news/articles/2001-09-30/tesco-bets-small-and-wins-big`.

[168] Robbins L, 1935, *An essay on the nature and significance of economic science*, 2$^{\text{nd}}$ Edition, MacMillan, London.

[169] Robusté F, Daganzo CF & Souleyrette RR, 1990, *Implementing vehicle routing models*, Transportation Research Part B: Methodological, **24(4)**, pp. 263–286.

[170] Saghiri S & Wilding R, 2016, *The journey toward omnichannel retailing*, Logistics and Transport Focus, December, pp. 30–32.

[171] Sargent RG, 2007, *Verification and validation of simulation models*, Proceedings of the 39th Winter Simulation Conference, Washington (DC), pp. 166–183.

[172] Saunders DR, Bex PJ & Woods RL, 2013, *Crowdsourcing a normative natural language dataset: A comparison of Amazon Mechanical Turk and in-lab data collection*, Journal of Medical Internet Research, **15(5)**, pp. 49–62.

[173] Savelsbergh M & Van Woensel T, 2016, *City logistics: Challenges and opportunities*, Transportation Science, **50(2)**, pp. 579–590.

[174] *Sharing cities: Activating the urban commons*, 2018, Shareable, Mountain View (CA).

[175] Shaw A, Horton J & Chen D, 2011, *Designing incentives for inexpert human raters*, Proceedings of the 2011 ACM Conference on Computer Supported Cooperative Work, Hangzhou, pp. 275–284.

[176] Shires JD & de Jong GC, 2009, *An international meta-analysis of values of travel time savings*, Evaluation and Program Planning, **32(4)**, pp. 315–325.

[177] Singer Y & Mittal M, 2013, *Pricing mechanisms for crowdsourcing markets*, Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, pp. 1157–1166.

[178] Slabinac M, 2015, *Innovative solutions for a "last-mile" delivery — A European experience*, Business Logistics in Modern Management, **15**, pp. 111–129.

[179] Small KA, 2012, *Valuation of travel time*, Economics of Transportation, **1(1)**, pp. 2–14.

[180] Small KA & Verhoef ET, 2007, *The economics of urban transportation*, Routledge, New York (NY).

[181] SmartCitiesWorld, 2019, *DHL launches its first regular urban drone delivery service*, [Online], [Cited May 2020], Available from `https://www.smartcitiesworld.net/news/dhl-launches-its-first-regular-urban-drone-delivery-service-4189`.

[182] Smedley T, 2013, *Can Patagonia and eBay make shopping more sustainable?*, [Online], [Cited July 2021], Available from `https://www.theguardian.com/sustainable-business/patagonia-ebay-making-shopping-more-sustainable`.

[183] Solomon MM, 1987, *Algorithms for the vehicle routing and scheduling problems with time window constraints*, Operations Research, **35(2)**, pp. 254–265.

[184] Soman D, 2001, *Effects of payment mechanism on spending behavior: The role of rehearsal and immediacy of payments*, Journal of Consumer Research, **27(4)**, pp. 460–474.

[185] Swaminathan JM & Tayur SR, 2003, *Models for supply chains in e-business*, Management Science, **49(10)**, pp. 1387–1406.

[186] Taou NS, 2021, Research Associate at *Centre for Sustainable Road Freight, Heriot-Watt University*, [Personal Communication], Contactable at `n.taou@hw.ac.uk`.

[187] Terzi N, 2011, *The impact of e-commerce on international trade and employment*, Proceedings of the 7^{th} International Strategic Management Conference, France, pp. 745–753.

[188] Threadless, 2021, *About us*, [Online], [Cited September 2021], Available from `https://www.threadless.com/about-us/`.

[189] Toth P & Vigo D, 2001, *The vehicle routing problem*, Society for Industrial and Applied Mathematics, Philadelphia (PA).

[190] Toth P & Vigo D, 2002, *Models, relaxations and exact approaches for the capacitated vehicle routing problem*, Discrete Applied Mathematics, **123(1-3)**, pp. 487–512.

[191] Toth P & Vigo D, 2014, *Vehicle Routing: Problems, methods, and applications*, Society for Industrial and Applied Mathematics, Phildelphia (PA).

[192] Tripathi A & Khazanchi D, 2016, *The value of crowdsourcing for complex problems: Comparative evidence from software developed by the crowd and professionals*, Journal of Computer Science Applications and Information Technology, **1(1)**, pp. 1–7.

[193] Turban DC, Outland J, King D, Lee JK & Liang TP, 2018, *Electronic commerce 2018: A managerial and social networks perspective*, 9^{th} Edition, Springer, Cham.

[194] Uber, 2021, *About us*, [Online], [Cited September 2021], Available from `https://www.uber.com/za/en/about/`.

[195] United States Environmental Protection Agency, 2021, *Inventory of U.S. greenhouse gas emissions and sinks: 1990–2019,* [Goverment Report 430-F-21-01], EPA, Washington (DC).

[196] Utomo DS, 2021, Research Associate at *Centre for Sustainable Road Freight, Heriot-Watt University*, [Personal Communication], Contactable at `d.utomo@hw.ac.uk`.

[197] Vakulenko Y, Shams P, Hellström D & Hjort K, 2019, *Service innovation in e-commerce last mile delivery: Mapping the e-customer journey*, Journal of Business Research, **101**, pp. 461–468.

[198] Van Rossum G & Drake FL, 2009, *Python 3 reference manual*, CreateSpace, Scotts Valley (CA).

[199] Veblen T, 1899, *The theory of the leisure class: An economic study of institutions*, Macmillan Company, New York (NY).

[200] Vidal T, Crainic TG, Gendreau M & Prins C, 2013, *Heuristics for multi-attribute vehicle routing problems: A survey and synthesis*, European Journal of Operational Research, **231(1)**, pp. 1–21.

[201] Vukovic M, 2009, *Crowdsourcing for enterprises*, Proceedings of the 2009 Congress on Services — Part I, Los Angeles (CA), pp. 686–692.

[202] Wang Y, Zhang D, Liu Q, Shen F & Lee LH, 2016, *Towards enhancing the last-mile delivery: An effective crowd-tasking model with scalable solutions*, Transportation Research Part E: Logistics and Transportation Review, **93**, pp. 279–293.

[203] Wardman M, 1988, *A comparison of revealed preference and stated preference models of travel behaviour*, Journal of Transport Economics and Policy, **22(1)**, pp. 71–91.

[204] Weber CL, Koomey JG & Matthews HS, 2010, *The energy and climate change implications of different music delivery methods*, Journal of Industrial Ecology, **14(5)**, pp. 754–769.

[205] Willard J, 1989, *Vehicle routing using r-optimal tabu search*, MSc Thesis, The Management School, Imperial College, London.

[206] Winston WL, 2004, *Operations research: Applications and algorithms*, 4$^{\text{th}}$ Edition, Brooks/Cole, Belmont (CA).

[207] Wolfe-Adam T, 2020, *AnyLogic-Pypeline*, [Online], [Cited November 2020], Available from `https://github.com/t-wolfeadam/AnyLogic-Pypeline`.

[208] Xing Y & Grant DB, 2006, *Developing a framework for measuring physical distribution service quality of multi-channel and "pure player" internet retailers*, International Journal of Retail and Distribution Management, **34(4-5)**, pp. 278–289.

[209] Yellow P, 1970, *A computational modification to the savings method of vehicle scheduling*, Operational Research Quarterly, **21(2)**, pp. 281–283.

[210] Zak PJ, 2008, *CouchSurfing 101*, [Online], [Cited January 2021], Available from `www.psychologytoday.com/blog/the-moral-molecule/200810/CouchSurfing-101`.

[211] Zhao Y & Zhu Q, 2014, *Evaluation on crowdsourcing research: Current status and future direction*, Information Systems Frontiers, **16(3)**, pp. 417–434.

APPENDIX A

Datasets

The datasets utilised were proposed as problem sets by Solomon [183], and are available online [140]. These problem sets include routing and scheduling environments and are generally used as a benchmark input data for the VRPTW. For the purposes of this thesis, however, time windows are not considered. Rather, the problem sets may be described by the number of customers to serve, the geographical distribution of these customers, as well as the delivery vehicle capacity. The available data are first distinguished with respect to the number of customers. Instances exist for the case of 25, 50, and 100 customers, respectively. Furthermore, the problem sets proposed by Solomon [183] may further be classified by three types of geographical distribution of customers. For problem instances R1 and R2, the customer geographical data are generated randomly by a random uniform distribution. Problem sets C1 and C2 denote the problem sets with clustered locations of customers, while problem sets RC1 and RC2 contain semi-clustered data. A semi-clustered problem refers to a combination of randomly generated data and clustered data. Finally, with respect to the delivery vehicle capacities, problem sets R1, C1, and RC1 have small delivery vehicle capacities, whereas problem sets R2, C2, and RC2 have large delivery vehicle capacities. Additional distinctions are made with respect to the scheduling horizons and customer time windows. As these aspects are not considered for the purpose of this thesis, no distinction is made between problem instances within a particular problem set for a particular number of customers.

APPENDIX B

Contents on a GitHub repository

This appendix details the primary contents of the GitHub repository titled "`ChristianMalan/customer_crowd_shipping`". This GitHub repository is made available in order to provide access to the proposed model for customer crowd-shipping, as well as the final thesis document. An overview of the primary files contained in the repository is provided subsequently.

Agent based simulation model. Access to the agent-based model is provided in the form of an AnyLogic script file. The file is titled `Customer_Crowd_Shipping_Model.alp`. The model is developed in AnyLogic version 8.7.2 and may be run on subsequent versions of the software. It is noted that for the agent-based simulation model to be executed on a personal computer, it is necessary to install the PypeLine custom library, as detailed by Adam-Wolfe [207].

Python **scripts.** Access is provided to the Python scripts utilised throughout the simulation execution through the use of the PypeLine custom library. The folder titled `vrp_algorithms` contains the files titled `savings.py` and `sweep.py`, which are executed in solving the CVRP formulated in §4.3.3.1, by the savings and sweep algorithms, respectively. Additionally, the files `calculate_cost2serve.py`, `OD_trip_evaluation.py`, and `capture_run_results.py` are contained in the repository and are critical to the execution of the simulation model.

Input data. Access is provided to the input data utilised in Chapter 5 and 6. Contained in the folder titled `data`, the Solomon [183] problem sets are made available in `json` format. The desired dataset may be selected as input to the simulation model through use of the GUI described in §4.7.

Thesis. In addition to the model, access to the final version of the written thesis is provided as an electronic copy titled `thesis.pdf`.

The GitHub repository may be accessed at:

`https://github.com/ChristianMalan/customer_crowd_shipping`