Two-Echelon Last-Mile Delivery Route Optimization with Autonomous Vehicles and Drones

Hamidreza Ensafian

BSc, MSc

A thesis submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy

Institute of Transport and Logistics Studies
The University of Sydney Business School
The University of Sydney
Australia

2023

In loving memory of my dear father, whose presence and support have been a source of inspiration throughout my academic pursuits. Though he is no longer with us, his unwavering belief in my abilities continues to guide me. I dedicate this thesis to his memory, cherishing the values he instilled in me and the lasting impact he has had on my life.

Abstract

This thesis focuses on investigating collaborative human-robot systems for delivery and pickup tasks. In Chapter 2, a novel Mixed Integer Linear Programming (MILP) formulation is proposed to model the problem, involving couriers serving customers while Autonomous Mobile Lockers (AMLs) assist them with deliveries and pickups. The MILP model considers a network of depots, stations, AMLs, and couriers, allowing couriers to perform multiple open routes and AMLs to perform single open routes. A heuristic algorithm is developed to solve large-scale instances, providing optimal solutions for some small-sized instances. Comparative analysis demonstrates the cost-saving potential of the proposed robot-human cooperative system.

Chapter 3 introduces a delivery system employing trucks and drones. A Mixed Integer Programming (MIP) model is presented, considering operational parameters and customer locations. Two efficient heuristic algorithms are developed to solve the vehicle routing problem for large instances, exhibiting good performance in terms of execution time and solution quality. Sensitivity analysis highlights the optimal ratio of flight range and load carrying capacity of drones based on network configurations and population densities.

In Chapter 4, a novel MIP model is proposed for e-grocery delivery, allowing customers to choose multiple time windows. The approach combines truck and drone deliveries with synchronization constraints at rendezvous points. A hybrid heuristic algorithm and Constraint Programming approach is developed for efficient solving. Experimental results show the algorithm's capability to provide quality solutions within seconds. Comparative analysis demonstrates the advantages of the proposed model over traditional delivery platforms. Sensitivity analysis on varied time window widths and customer preferences reveals the diminishing effect on the objective as the proportion

of customers choosing multiple time slots increases. Encouraging customers to select multiple time slots enhances customer service and enables tighter time window management.

Overall, this thesis presents innovative models and algorithms for collaborative human-robot delivery systems, showcasing their efficiency, cost-saving potential, and adaptability to different network configurations and customer preferences.

Statement of Original Authorship

I hereby declare that the content of this thesis is solely the result of my own research and efforts. This work has not been previously submitted for any degree or other purposes. I affirm that all sources utilized in the preparation of this thesis have been appropriately acknowledged, and any assistance received during the process has been duly recognized.

Acknowledgements

First and foremost, I am immensely grateful to Professor Michael Bell for his guidance, expertise, and unwavering support as my supervisor. His invaluable insights and constructive feedback have been pivotal in shaping the direction and quality of this research.

I would also like to extend my heartfelt appreciation to Dr. Jyotirmoyee Bhattacharjya for her co-supervision and valuable contributions throughout this journey. Her guidance and scholarly advice have been invaluable in enriching the content and strengthening the overall coherence of this thesis.

I would like to acknowledge the immense contribution of Professor D. Glenn Geers, who generously provided his time and expertise in editing and reviewing the manuscript. His valuable suggestions and inputs greatly enhanced the clarity and coherence of the thesis.

I extend my sincere thanks to Dr. Philip Kilby for his assistance in solving methodological challenges. His expertise and insightful discussions were instrumental in overcoming obstacles and refining the research approach.

I would like to express my gratitude to Dr. Jun Li for her support in conceptualizing Chapter 1. Her expertise in the subject matter and valuable discussions have been invaluable in framing the research objectives and establishing a solid foundation for the thesis.

Special thanks go to Abdullah Zare Andaryan for his invaluable assistance in coding and methodology. His dedication, technical expertise, and collaborative spirit greatly contributed to the successful execution of the research project.

I would like to extend my sincere gratitude to the faculty members of the Institute of Transport and Logistics Studies (ITLS) for their invaluable support and contributions throughout this research endeavor. In particular, I would like to express my deepest appreciation to Professor David Hensher, who holds the esteemed position of Chair at ITLS. His guidance, mentorship, and expertise have been invaluable in shaping my understanding of the field and providing valuable insights that have significantly enriched this thesis.

I would also like to acknowledge and thank all the other esteemed faculty members at ITLS for their valuable teachings, academic rigor, and dedication to fostering an environment of intellectual growth and scholarly excellence. Their collective wisdom and commitment to advancing the field of transport and logistics studies have been instrumental in shaping my research and academic journey.

Furthermore, I would like to express my gratitude to the entire administrative and support staff at ITLS for their assistance and logistical support throughout my studies. Their professionalism and dedication to ensuring a conducive research environment have been greatly appreciated.

Finally, I would like to thank all the faculty members, research colleagues, and friends who have provided encouragement, support, and inspiration throughout this endeavor. Their intellectual exchanges and moral support have been instrumental in shaping this thesis.

I am indebted to each of these individuals for their invaluable contributions, and I express my deepest appreciation for their guidance, support, and encouragement throughout this research journey.

Publications

The following papers were published during the course of my doctoral candidature. Papers directly related to this thesis topic are marked with an asterisk (∗) at the end.

Journal Articles

- Li, J., **Ensafian, H**., Bell, M. G., & Geers, D. G. (2021). Deploying autonomous mobile lockers in a two-echelon parcel operation. *Transportation Research Part C: Emerging Technologies, 128*, 103155.
- **Ensafian, H**., Andaryan, A. Z., Bell, M. G., Geers, D. G., Kilby, P., & Li, J. (2023). Cost-optimal deployment of autonomous mobile lockers co-operating with couriers for simultaneous pickup and delivery operations. *Transportation Research Part C: Emerging Technologies, 146*, 103958.∗

Conferences

- Li, J., **Ensafian, H**., Bell, M. G., & Geers, D. G. (2022). Deploying autonomous mobile lockers in a two-echelon parcel operation. *24th International Symposium on Transportation and Traffic Theory, ISTTT24.*
- Li, J., Bell, M. G**., Ensafian, H**. (2021). Saving cost by the deployment of Autonomous Mobile Lockers (AMLs) for last-mile deliveries. *8TH INTERNATIONAL SYMPOSIUM ON TRANSPORT NETWORK RELIABILITY, INSTR 2021.*
- Li, J., **Ensafian, H**., Bell, M. G., & Geers, D. G. (2021). Two-Echelon Open Vehicle Routing Problem with Autonomous Mobile Lockers. *22nd Conference of the International Federation of Operational Research Societies, IFORS 2021.*

Table of Contents

Table of Figures

List of Tables

List of Abbreviations

Abbreviation	Full name
FSTSP	Flying Sidekick Travelling Salesman Problem
PDSTSP	Parallel Drone Scheduling Travelling Salesman Problem
VRP	Vehicle Routing Problem
VRPD	Vehicle Routing Problem with Drones
OVRP	Open Vehicle Routing Problem
MVRP	Multiple Vehicle Routing Problem
MVRPD	Multi-visit Vehicle Routing Problem with Drones
VRPTW	Vehicle Routing Problem with Time Window
VRPMTW	Vehicle Routing Problem with Multiple Time Window
VRPDMTW	Vehicle-drone Routing Problem with Multiple Time Window
AGV	Autonomous Ground Vehicle
UAV	Unmanned Arial Vehicle
AML	Autonomous Mobile Locker
2E-VRP	2-Echelon Vehicle Routing Problem
CP	Constraint Programming
ALNS	Adaptive Large Neighbourhood Search
MILP	Mixed-integer Linear Programming
MILNP	Mixed-integer Non-linear Programming
SA	Simulated Annealing
BTA	Backtracking Algorithm
ABSM	Adaptive Backtracking-SA Algorithm
CWH	Clarke and Wright Heuristic
BCWH	Backward Clarke and Wright Heuristic
BFCWH	Backward-forward Clarke and Wright Heuristic

Chapter 1: Emerging technologies in last-mile delivery: An overview

1.1 INTRODUCTION

The rise of e-commerce has revolutionized the retail industry, and online shopping has become a popular option for consumers. This trend has been particularly evident, where the convenience of online shopping and home delivery has become increasingly important for time-pressed consumers.

According to a report by Statista (Marina Pasquali, 2023) - as an example - the global online food delivery market is anticipated to experience a growth rate of 12.67% between 2023 and 2027, leading to a market size of approximately US$1.65 trillion by 2027. This report states that grocery delivery segment is forecasted to generate approximately US$8.25 billion in revenue by 2023. It is anticipated that the revenue will experience a compound annual growth rate (CAGR) of 12.20% from 2023 to 2027. As a result, the market volume is projected to reach around US$13.08 billion by 2027. This has prompted supermarkets to invest heavily in online infrastructure, including web-based commerce platforms and delivery networks (*Making Online Grocery a Winning Proposition | McKinsey*, n.d.). In the United States, major retailers such as Walmart, Target, and Amazon have all expanded their online grocery offerings, while in the UK, Tesco and Sainsbury's have introduced home delivery and click-and-collect options.

The COVID-19 pandemic has further accelerated this growth, as consumers around the world have been forced to adopt new behaviours and practices to minimize the risk of infection. With social distancing measures in place, many physical stores were forced to close, and consumers were left with limited options for shopping. As a result, more and more people turned to online shopping to fulfil their needs, leading to a surge in demand for e-commerce platforms (Perrotta et al., 2021).

However, the pandemic also highlighted some of the challenges and limitations of online shopping, including the logistical and operational issues associated with delivery and fulfilment. As the volume of online orders surged, many retailers struggled to keep up with the demand, leading to delays and disruptions in delivery times (Loske, 2020). Furthermore, the increased reliance on delivery services also raised concerns about sustainability and environmental impact, as more delivery vehicles hit the roads to fulfil orders.

Moreover, the convenience of online shopping has also contributed to the growth of the online market. Customers can easily browse and purchase products from the comfort of their own homes, eliminating the need for a trip to the shopping centres. This is especially appealing to busy individuals and families, as well as those with mobility issues or living in remote areas (Hasan & Rahim, 2008).

The rise of online shopping has been driven by various factors, one of which is the growing demand for same-day and next-day delivery options. Consumers today expect faster and more convenient delivery services, and they are willing to pay a premium for it. According to a report published by (*Same-Day Delivery: Ready for a New Omnichannel Strategy | McKinsey*, n.d.), same-day and next-day delivery options are becoming increasingly important in the online market. The report found that consumers are willing to pay up to 10% more for same-day delivery and up to 5% more for next-day delivery. In addition, retailers who offer these delivery options can increase their market share by up to 5%. Same-day and next-day delivery options can be especially attractive for customers who need to purchase items quickly or who may have forgotten to buy something earlier.

To meet the growing demand for faster delivery options, many companies have invested in delivery networks that can offer same-day or next-day delivery. One approach that has gained popularity in recent years is the use of hybrid delivery networks that utilize autonomous vehicles (AV), robots, and drones to deliver items to customers (Chung, 2021). These networks allow retailers to offer faster delivery times by utilizing automated vehicles to deliver items quickly and efficiently, while still relying on couriers and traditional vehicles, such as trucks and vans, for longer distance deliveries.

Hybrid delivery networks can be especially effective in urban areas, where traffic congestion can cause delays in delivery times (Samouh et al., 2020). For example, by using drones to deliver items directly to customers, retailers can avoid traffic and other obstacles that can slow down traditional delivery methods. This can help to improve customer satisfaction and loyalty, as customers appreciate the convenience and speed of these delivery options.

One example of a company that has implemented a hybrid delivery network is Amazon. In 2013, the company announced its Prime Air program, which utilizes drones to deliver packages to customers within 30 minutes (Shavarani et al., 2018). While the program has faced regulatory and logistical challenges, it highlights the potential of using drones for delivery purposes. Other companies, such as Walmart and Kroger, have also invested in drone delivery programs in an effort to offer faster and more convenient delivery options to their customers. Other companies, such as UPS and FedEx, have also explored the use of drones for package delivery (Welch, 2015). However, the use of drones and trucks in a hybrid delivery network is a relatively new concept that has the potential to revolutionize the online markets.

This study explores the use of a delivery system that utilizes a network consisting of a depot and a fleet of drones/autonomous vehicles and trucks to deliver items to customers. This is an innovative approach that has the potential to significantly reduce delivery times and transportation cost while increasing customer satisfaction.

A key advantage of a hybrid delivery network is its ability to reach remote and hard-to-access areas. For example, drones can be used to deliver items to customers living in rural or mountainous regions where traditional delivery methods may be impractical or too expensive. Additionally, the use of autonomous vehicles and drones can help reduce the carbon footprint of delivery services by reducing the number of traditional vehicles on the road, which in turn reduces emissions and traffic congestion.

Section 1.2 provides a systematic review that investigates studies focusing on various models of the Vehicle Routing Problem (VRP) which incorporate autonomous vehicles, delivery robots, and drones.

1.2 SYSTEMATIC REVIEW

This section examines the utilization of autonomous vehicles, drones, and delivery robots in last-mile logistics, covering both pickup and delivery operations. The review is based on the concept-centric and Systematic Review method, as described by (Cheung et al., 2021; J. Li, Rombaut, et al., 2021; Mualla et al., 2019). The systematic literature review process is outlined in Figure 1.1.

In this analysis, the focus is on the different Vehicle Routing Problem (VRP) models and objectives utilized in previous studies. To determine the most relevant articles for our review, we established selection criteria and utilized the Scopus database to search for English language journal articles related to autonomous vehicles, drones, or delivery robots in logistics. This search was narrowed by using specific keywords such as "autonomous vehicle", "drone", "delivery robot", "routing problem", and "logistics". Additionally, the keywords "pick or deliver", "last-mile", and "last mile" are included to further refine our search. This search was limited to English language articles from journals and had the following syntax in Scopus: "autonomous vehicle" OR "drone" OR "delivery robot" AND "routing problem" AND "logistics" OR "pick or deliver" OR "last-mile" OR "last mile" AND LANGUAGE=English AND DOCTYPE=ar AND SRCTYPE=j.

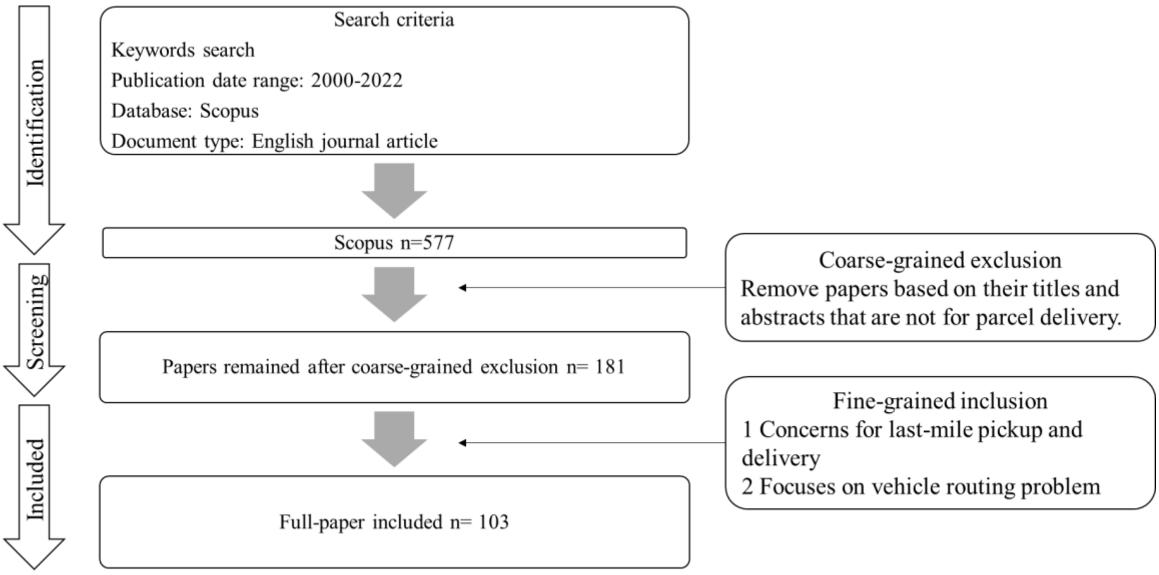


Figure 1.1. Flow chart of the systematic review process.

Out of 577 English language journal articles, a selection was made, and a keyword visualization was generated using VOSviewer on the articles sourced from the Scopus database. The size of the dot represents the frequency of the relevant keywords, as shown in Figure 1.2. The visualisation highlights that “drone” is the most frequently used keyword, represented by the blue cluster. The second most commonly used keyword, represented by the red cluster, is “autonomous vehicle”. However, the research on autonomous vehicles mainly focuses on transportation issues such as transportation systems, traffic congestion, roads, and streets. The green cluster represents delivery robot research, where the dominant keywords are integer programming and trucks. The research on drones and delivery robots also needs to account for the use of trucks as they are often utilised together.

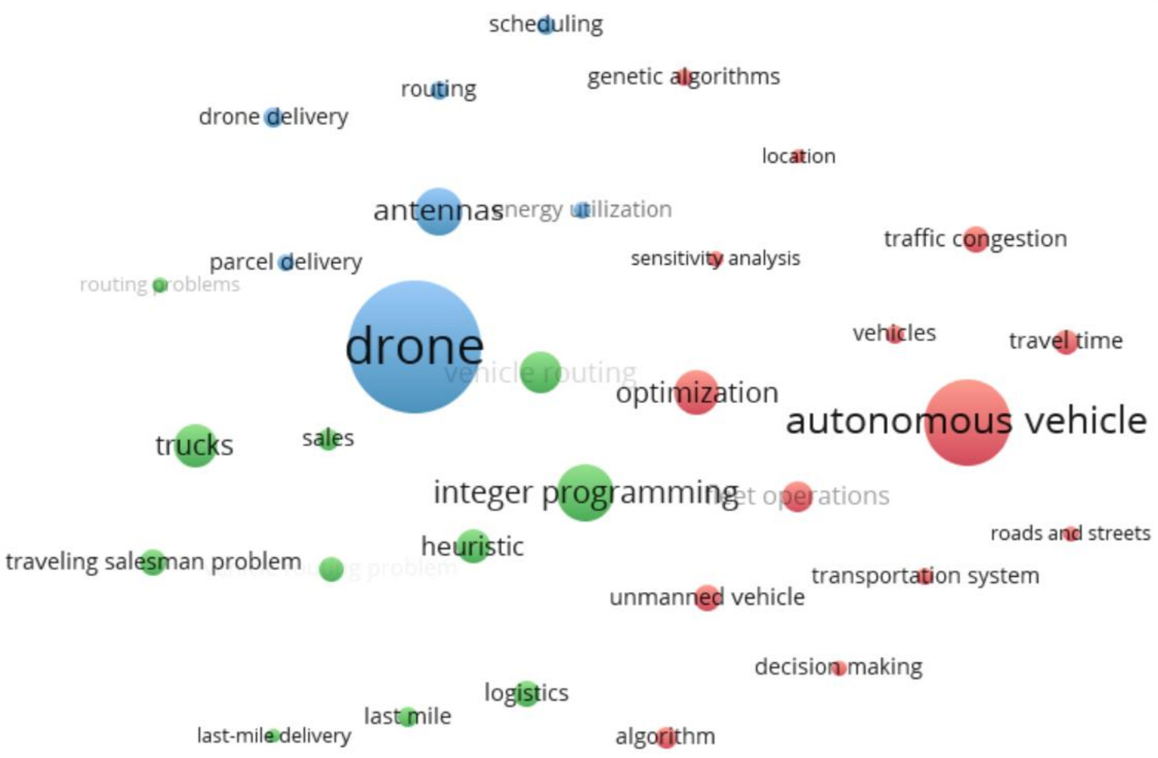


Figure 1.1. Keyword visualisation for VRP with autonomous vehicle, drone, and delivery robot for last-mile pickup and delivery

The process of filtering through titles and abstracts led to the elimination of articles that were not relevant to parcel delivery, resulting in a final count of 181 articles. Upon closer examination of the full text of each article, an even smaller set of 103 articles was identified as relevant to the topic of last-mile pickup and delivery addressing VRP. These articles were found to have been published between 2002 and 2022, with the exception of review papers. The publication year of the selected articles is summarised in Figure 1.3, which indicates a significant increase in the number of VRP articles related to last-mile pickup and delivery after 2018. The majority of the articles focused on the application of drones in parcel delivery, while the usage of delivery robots is gradually gaining popularity each year.

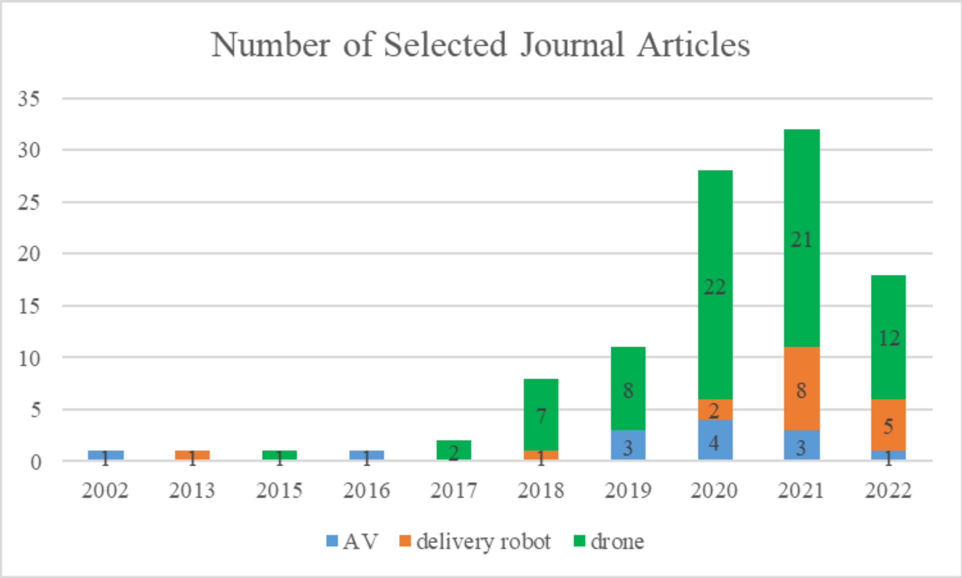


Figure 1.2. Summary of the number of selected journal articles published since 2002.

Table 1.1 highlights the leading journals that have published articles on various aspects of AVs, drones, and delivery robots for last-mile pickup and delivery. Articles related to AVs have been published most frequently in Transportation Research Part E, followed closely by Computers and Operations Research. For delivery robots, the European Journal of Operational Research leads the way, with Transportation Research Part E and Networks following closely. Articles on drones have been published in Transportation Research Part C most frequently, with other notable journals including Networks, IEEE Access, Computers and Operations Research, and European Journal of Operational Research. It is evident from Table 1.1 that the application of emerging technologies in last-mile pickup and delivery is an interdisciplinary field, encompassing engineering, computer science, transportation, and operations research.

Table 1.1. A breakdown of three type of technologies in which the selected articles appeared.

Technology type	Journal name	Number
AV	Transportation Research Part E	3
	Computers and Operations Research	2
	Others*	8
Delivery Robot	European Journal of Operational Research	6
	Transportation Research Part E	3
	Networks	2

	Others*	6
Drone	Transportation Research Part C	16
	Networks	8
	IEEE access	7
	Computers and Operations Research	5
	European Journal of Operational Research	5
	Applied sciences	3
	Transportation Research Part B	3
	Computers & industrial engineering	2
	International transactions in operational research	2
	Transportation Research Part E	2
	Others*	20

*: "Others" represents the subcategory with less than two articles.

1.3 CLASSIFICATION OF THE OPTIMISATION OBJECTIVES

Of the 103 articles selected for this review, the majority, 95, focused on minimisation objectives such as reducing time, cost, the number of facilities, or distance. Conversely, only 11 articles were dedicated to maximising objectives like maximising successful deliveries or profits. Table 1.2 categorises these objectives into two groups: minimisation and maximisation objectives. The review included studies that aimed to minimise fleet size (Han et al., 2020; Troudi et al., 2018; S. Yu et al., 2020), hubs (Bakach et al., 2021), and stations (Pinto & Lagorio, 2022), which were therefore grouped into the category of minimising facilities. Additionally, articles on minimising energy consumption (Coelho et al., 2017; Iranmanesh et al., 2020) and emissions (Elsayed & Mohamed, 2020; Rashidzadeh et al., 2021) were grouped under minimising environmental impact. Furthermore, minimisation of penalty (Choudhury et al., 2022; Liu et al., 2021) was considered as a type of cost reduction. Thus, in the following sections, we will focus on the categorisation of studies that minimise time, cost, facilities, distance, and environmental impact, as well as maximising successful delivery.

Table 1.2. Distribution of articles based on research objective.

Category	Objectives (#) *	Selected articles
Minimization	Time (43)	(Almuhaideb et al., 2021; Bai et al., 2020; Boccia et al., 2021; Boysen, Briskorn, et al., 2018; Cavani et al., 2021; C. Chen, Demir, & Huang, 2021; C. Chen, Demir, Huang, et al., 2021;

	Coelho et al., 2017; Dell'Amico et al., 2021b, 2021a; Dorling et al., 2017; Freitas & Penna, 2020; Ghelichi et al., 2021; Gomez-Lagos et al., 2021; Gonzalez-R et al., 2020; Ham, 2018; Jeon et al., 2021; Kitjacharoenchai et al., 2020; Leon-Blanco et al., 2022; Lin et al., 2020; H. Luo et al., 2019; Z. Luo et al., 2021; Masone et al., 2022; Mbiadou Saleu et al., 2018, 2022; Momeni et al., 2022; Moshref-Javadi et al., 2021; Moshref-Javadi, Hemmati, et al., 2020; Moshref-Javadi, Lee, et al., 2020; Murray & Chu, 2015; Murray & Raj, 2020; Peng et al., 2019; Phan & Suzuki, 2016; Pina-Pardo et al., 2021; Poikonen & Golden, 2020; Raj & Murray, 2020; Reed et al., 2022; Schermer et al., 2019, 2020; Singgih et al., 2020; Tamke & Buscher, 2021; Vu et al., 2022; K. Wang et al., 2022)
Cost (38)	(Bakach et al., 2021; Campbell et al., 2018; Cheng et al., 2020; Choudhury et al., 2022; Coindreau et al., 2021; Cokyasar et al., 2021; Dang et al., 2021; Das et al., 2021; Di Puglia Pugliese et al., 2021; Dorling et al., 2017; Ha et al., 2018; Hernández et al., 2020; H. Huang et al., 2020; S. H. Huang et al., 2022; Karak & Abdelghany, 2019; H. Li et al., 2022; Jun Li et al., 2021; Liu et al., 2020; Liu, Deng, et al., 2021; Liu, Yan, et al., 2021; Momeni et al., 2022; Mourad et al., 2021; Nguyen et al., 2022; Ostermeier et al., 2022; Peng et al., 2019; Rashidzadeh et al., 2021; Resat, 2020; Sacramento et al., 2019; Scherr et al., 2019, 2020; Simoni et al., 2020; Ulmer & Streng, 2019; Z. Wang & Sheu, 2019; Xia et al., 2021; S. Yu et al., 2020, 2022; M. Zhang & Yang, 2022)
Distance (11)	(Cai et al., 2021; Coelho et al., 2017; Gu & Wallace, 2021; Kaspi et al., 2002; Phan & Suzuki, 2016; Pinto & Lagorio, 2022; Poikonen & Golden, 2020; Puerto & Valverde, 2022; Troudi et al., 2018; J. J. Q. Yu, 2019; M. Zhang & Yang, 2022)
Facility (10)	(Al-Turjman et al., 2019; Bakach et al., 2021; Cai et al., 2021; Coelho et al., 2017; Han et al., 2020; Phan & Suzuki, 2016; Pinto & Lagorio, 2022; Schwerdfeger & Boysen, 2020; Troudi et al., 2018; S. Yu et al., 2020)
Environment impact (9)	(Coelho et al., 2017; Elsayed & Mohamed, 2020; Han et al., 2020; Iranmanesh et al., 2020; Liu, Deng, et al., 2021; Liu,

		Yan, et al., 2021; Rashidzadeh et al., 2021; Ren et al., 2022; Resat, 2020)
	Tardiness (3)	(Alfandari et al., 2021; Boysen, Schwerdfeger, et al., 2018; Jun et al., 2021)
	Flow (1)	(Hamzeei et al., 2013)
Maximization	Satisfied demand (8)	(Chauhan et al., 2019; X. Chen et al., 2022; Dayarian et al., 2020; Hong et al., 2018; Jung & Kim, 2022; Kim et al., 2020; Phan & Suzuki, 2016; Rashidzadeh et al., 2021)
	Profit (2)	(Shui & Li, 2020; G. Zhang et al., 2021)
	Speed (1)	(Coelho et al., 2017)

*: The numbers in brackets indicate the number of articles involving that objective.

1.3.1 Time minimization

According to Table 1.2, the main objective of a majority of the articles (39) is to reduce the delivery time. Table 1.3 provides a more detailed categorization of the studies. Out of the selected articles, 77% (30 articles) were centered on reducing the completion time.

The objective functions in some studies are designed to minimize the waiting time of vehicles (Masone et al., 2022; Phan & Suzuki, 2016) and vehicle travel time (Boccia et al., 2021; Dell'Amico et al., 2021b; Gomez-Lagos et al., 2021; M. Lin et al., 2020). The service time is also considered in these studies, as seen in the research by (C. Chen, Demir, & Huang, 2021), who examine the delivery robot and include the vehicle travel time, vehicle waiting time, and service time in their objective function.

Three articles just focused on minimizing the customers' waiting time. (Moshref-Javadi et al., 2021) proposed minimising the sum of arrival times at customer locations, which can also be seen as minimising the customer waiting time (Moshref-Javadi, Hemmati, et al., 2020; Moshref-Javadi, Lee, et al., 2020). Unlike previous studies, they do not take into account the return time to the depot.

Six articles considered the minimization of makespan as their objective function. Minimizing the makespan refers to minimizing the total time needed to complete all tasks, including transportation time between locations, loading and unloading times, and any other non-service times. This means that the objective is to minimize the time it takes for all tasks to be completed, which can result in fewer vehicles being used or shorter delivery windows. Minimizing completion time, on the other hand, refers to minimizing the time required to complete each individual task or activity, such as delivering a package to a specific location (Bräysy & Gendreau, 2005; Cordeau et al., 2002; Laporte, 1992; Solomon, 1987). This objective function focuses on optimizing the time required for each task, rather than the overall time required for all tasks.

The choice between these two objective functions depends on the specific goals and constraints of the transportation problem being solved. Minimizing the makespan may be appropriate when there are strict time constraints or when the focus is on minimizing the overall transportation time, while minimizing completion time may be more appropriate when the focus is on improving service levels or reducing waiting times at specific locations.

The majority of the selected papers focused on the drone-truck routing problem, which involves the integration of drones, also known as Unmanned Aerial Vehicles (UAVs), into the traditional truck routing problem. By analysing the objective functions, it is observed that several articles (Dell'Amico et al., 2021a; Jeon et al., 2021; Z. Luo et al., 2021a; Murray & Raj, 2020; Pina-Pardo et al., 2021; Schermer et al., 2020; Tamke & Buscher, 2021a) have developed mixed integer linear programming (MILP) models based on (Murray & Chu, 2015) model for drone delivery. The initial model related to the Flying Sidekick Travelling Salesman Problem (FSTSP) and Parallel Drone Scheduling TSP (PDSTSP) was presented by (Murray & Chu, 2015). The FSTSP is designed for one vehicle and one drone delivery scenario (Murray & Chu, 2015a). Later, (Murray & Raj, 2020) introduced the multiple flying sidekicks travelling salesman problem (mFSTSP) which considered a vehicle equipped with a fleet of drones delivering parcels to customers. (Jeon et al., 2021) further expanded on this by including the pickup demand customer nodes in their model, allowing for the serving of pickup demands after completing the delivery tasks.

In recent studies, various factors have been considered when launching and retrieving drones. The service time (Z. Luo et al., 2021; Murray & Raj, 2020; Tamke & Buscher, 2021) and loading/unloading time (Jeon et al., 2021; Pina-Pardo et al., 2021) have been included by some researchers to account for the total time spent on a mission. However, other studies (Dorling et al., 2017; Kitjacharoenchai et al., 2020a; Mbiadou Saleu et al., 2018, 2022; Singgih et al., 2020) prioritise the completion time of each mission over the return time of the vehicle and drone to the depot.

While most studies consider the return of the vehicles and drones to the depot, a few (Bai et al., 2020; Gonzalez-R et al., 2020; Leon-Blanco et al., 2022) only consider the time to reach the final customer.

While the time window is a critical aspect of home delivery operations, only a limited number of studies (3 articles) have integrated it into their models. Table 1.3 also shows that mixed-integer linear programming is the most frequently used approach in this research area (26 articles). Given the complicated nature of VRP models, most studies rely on heuristic or metaheuristic algorithms, which can provide near-optimal solutions but do not guarantee optimal solutions. Only a handful of studies (7 articles) have devised exact algorithms capable of producing optimal solutions. The Branch & Cut algorithm is the most widely used among the exact algorithms developed for this purpose.

Table 1.2. Distribution of articles based on research objective in minimizing time.

Selected articles	P / D	Travel Time					TW	Obj (min)	Model	Solving Approach
Authors (year)		V	AV	UAV	R	H				
(Almuhaideb et al., 2021)	D	√		√				CT	NA	Metaheuristics
(Bai et al., 2020)	D	√		√				CT	NA	Heuristics combined
(Boccia et al., 2021)	D	√		√				CT	NA	Branch-and-Cut and column generation
(Boysen, Briskorn, et al., 2018)	D	√			√			Makespan	MILP	Heuristic
(Cavani et al., 2021)	D	√		√				CT	MILP	Branch-and-Cut
(C. Chen, Demir, & Huang, 2021)	D	√			√		√	CT	MILP	Heuristic
(C. Chen, Demir, Huang, et al., 2021)	D	√			√		√	CT	MILP	Metaheuristics
(Coelho et al., 2017)	P&D	√		√				CT	MILP	Metaheuristics
(Dell'Amico et al., 2021a)	D	√		√				CT	MILP	NA

(Dell’Amico et al., 2021b)	D	√	√			CT	MILP	NA
(Dorling et al., 2017)	D	√	√			CT	NA	Heuristic
(Freitas & Penna, 2020)	D	√	√			CT	NA	Combined Heuristic-exact algorithm
(Ghelichi et al., 2021)	D		√			CT	MILP	Heuristic
(Gomez-Lagos et al., 2021)	D	√	√			Makespan	MILP	Metaheuristics
(Gonzalez-R et al., 2020)	D	√	√			CT	MILP	Heuristic
(Ham, 2018)	P&D	√	√		√	Makespan	CP	NA
(Jeon et al., 2021)	P&D	√	√			CT	MILP	Heuristic
(Kitjacharoenchai et al., 2020a)	D	√	√			CT	MILP	Heuristic
(Leon-Blanco et al., 2022)	D	√	√			CT	NA	Agent-based approach
(Lin et al., 2020)	D	√	√			CT	MILP	Heuristic
(Z. Luo et al., 2021)	D	√	√			Makespan	MILP	Heuristic
(Masone et al., 2022)	D	√	√			CT	NA	2-stage iterative algorithm
(Mbiadou Saleu et al., 2018)	D	√	√			CT	NA	2-stage Heuristic
(Mbiadou Saleu et al., 2022)	D	√	√			CT	MILP	Hybrid metaheuristic
(Momeni et al., 2022)	D	√	√			CT /Cost	MILP	Benders’ decomposition
(Moshref-Javadi, Lee, et al., 2020)	D	√	√			Customer waiting time	MILP	Hybrid heuristic
(Moshref-Javadi, Hemmati, et al., 2020)	D	√	√			Customer waiting time	MILP	Heuristic

(Moshref-Javadi et al., 2021)	D	√		√			Customer waiting time	MILP	Heuristic
(Murray & Chu, 2015)	D	√		√			CT	MILP	Heuristic
(Murray & Raj, 2020)	D	√		√			CT	MILP	Heuristic
(Peng et al., 2019)	D	√		√			CT	NA	Hybrid metaheuristic
(Pina-Pardo et al., 2021)	D	√		√			CT	MILP	Decomposition algorithm
(Poikonen & Golden, 2020)	D	√		√			CT	NA	Heuristic
(Raj & Murray, 2020)	D	√		√			Makespan /Cost	NA	Heuristic
(Reed et al., 2022)	D		√			√			
(Schermer et al., 2019a)	D	√		√			Makespan	MILP	Metaheuristic
(Schermer et al., 2020)	D	√		√			CT	MILP	Branch-and-Cut
(Tamke & Buscher, 2021)	D	√		√			CT	MILP	Branch-and-Cut
(Vu et al., 2022)	D	√		√			CT	MILP	Metaheuristic
(K. Wang et al., 2022)	D	√		√			CT	DP	Greedy Algorithm

P: Pickup; D: Delivery; CT: Completion Time; MILP: Mixed-integer Linear Programming
V: Traditional Vehicles; AV: Autonomous Vehicles; UAV: Unmanned Arial Vehicles (Drone); R: Robot; H: Human; NA: Not Available.

1.4 COST MINIMIZATION

The objective function in a majority of the articles (32) focuses on minimising travel costs, as it is a crucial aspect of company competitiveness (Mesjasz-Lech & Włodarczyk, 2022). These articles mainly focus on minimising the cost of vehicle and robot travel, or vehicle and drone travel. However, the research on the use of drones, robots, and autonomous vehicles is still in the theoretical modelling stage, lacking actual case data to verify their effectiveness (Resat, 2020). Minimising fixed costs (with 12 articles) is the second dominant research focus, as the cost of sites and facilities is substantial and not easily changeable. This includes minimising the cost of purchasing or renting depots, satellites, vehicles, drones, and battery-swapping machines (Cokyasar et al., 2021; H. Huang et al., 2020; Li et al., 2021; Liu, Deng, et al., 2021; Liu, Yan, et al., 2021; Rashidzadeh et al., 2021; Salama & Srinivas, 2020).

Another dominant research focus in this field is on reducing operation costs, as evidenced by nine articles on the topic. The second most common focus is on minimising penalties, which is covered in eight articles. In terms of minimisation of penalties, studies have examined the cost of delays (Dang et al., 2021; Ostermeier et al., 2022; M. Zhang & Yang, 2022), early and late arrivals (Das et al., 2021; Liu, Yan, et al., 2021), unsuccessful tasks (Choudhury et al., 2022), and waiting time of vehicles or drones (Ha, Deville, Pham, & Hà, 2018; H. Li et al., 2022). In order to reduce operation costs, researchers have looked at minimising handling charges at depots and satellites (Liu, Deng, et al., 2021; Liu, Yan, et al., 2021; Simoni et al., 2020; Ulmer & Streng, 2019), as well as reducing inventory holding costs for drones (Resat, 2020), drone dispatch and collection costs (Karak & Abdelghany, 2019), battery swapping costs for drones (Cokyasar et al., 2021), and operating costs for robots (Bakach et al., 2021; Mourad et al., 2021). Among the remaining eight articles, researchers have also looked at reducing drone energy costs (Cheng et al., 2020; Dorling et al., 2017; Xia et al., 2021), labour costs (Coindreau et al., 2021; Hernández et al., 2020; Li et al., 2021), goods transport costs (Scherr et al., 2019), and outsourcing costs (Scherr et al., 2020).

Table 1.4 shows that Mixed-Integer Linear Programming (MILP) is the most frequently used type of mathematical model in the literature, followed by a few studies that employed Dynamic Programming (DP) or Mixed-Integer Non-Linear Programming (MINLP). Similar to the studies aiming to minimize time, most of the research in this field also employed heuristic or metaheuristic algorithms to address the problem. However, only three studies presented exact algorithms to obtain optimal solutions. Furthermore, 12 studies incorporated time windows into their models, although this crucial aspect of home delivery operations adds complexity to the problem.

Table 1.3. Distribution of articles based on research objective in minimizing cost.

Selected articles	P / D	Travel cost					Fixed cost		Operation	Penalty	Energy	TW	Model	Solving Approach
Authors (year)		Vehicle	AV	UAV	Robot	H	Facility	Fleet	cost		consumption			
(Bakach et al., 2021)	D	√			√				√			√	MILP	NA
(Cheng et al., 2020)	D			√									NLP	Branch-and-cut algorithm
(Coindreau et al., 2021)	D	√		√				√	√		√	√	MILP	NA
(Cokyasar et al., 2021)	D	√		√			√	√	√				MILP	Exact
(Dang et al., 2021)	P&D		√							√		√	MILP	Heuristic
(Das et al., 2021)	D	√		√						√		√	MILP	Hybrid metaheuristic
(Di Puglia Pugliese et al., 2021)	D	√		√								√	MILP	Heuristic
(Dorling et al., 2017)	D	√		√				√		√	√	√	MILP	Heuristic
(Ha, Deville, Pham, & Hà, 2018)	D	√		√						√			MILP	Heuristic/Excat
(Hernández et al., 2020)	D	√		√									MILP	NA
(S. H. Huang et al., 2022)	D	√		√				√					MILP	Metaheuristic

(Karak & Abdelghany, 2019a)	P&D	√		√					√			MILP	Heuristic
(H. Li et al., 2022)	D	√		√						√	√	MILP	Heuristic
(Jun Li et al., 2021)	P&D		√			√	√	√				MILP	Heuristic
(Liu et al., 2020)	D	√		√								NLP	Heuristic
(Liu, Deng, et al., 2021)	D	√			√		√	√	√			MILP	Hybrid Heuristic
(Liu, Yan, et al., 2021)	D	√			√		√	√	√	√	√	MILP	Hybrid Heuristic
(Mourad et al., 2021)	P&D				√				√		√	MILP	Heuristic
(Nguyen et al., 2022)	D	√		√								MILP	Heuristic
(Ostermeier et al., 2022)	D	√			√					√	√	MILP	Heuristic
(Peng et al., 2019)	D	√		√									Hybrid metaheuristic
(Rashidzadeh et al., 2021)	D			√			√	√				MILP	NA
(Resat, 2020)	D	√		√			√		√		√	MILP	NA
(Sacramento et al., 2019)	D	√		√								MILP	Heuristic
(Salama & Srinivas, 2020)	D	√		√				√				MILP	Heuristic
(Scherr et al., 2019)	D	√	√				√					MILP	NA

(Simoni et al., 2020)	D	√			√		√			MILP	Heuristic
(Ulmer & Streng, 2019)	D		√				√			DP	NA
(Z. Wang & Sheu, 2019)	D	√		√		√				MILP	Branch-and-price algorithm
(S. Yu et al., 2020)	D		√					√		MINLP	Heuristic
(S. Yu et al., 2022)	P&D	√		√					√	MILP	Hybrid metaheuristic

P / D: pickup or delivery; P&D: pickup and delivery; NA: Not Available.

1.5 ENVIRONMENTAL IMPACT

There are three primary types of research aimed at reducing environmental impact. The first type focuses on reducing CO_2 emissions (Elsayed & Mohamed, 2020; Rashidzadeh et al., 2021; Resat, 2020) by calculating the amount of CO_2 generated in power generation facilities to meet the drone's power needs. Factors such as the drone body mass, battery mass, parcel mass, gravitational constant, drag coefficient, and total power transfer efficiency are considered in the calculation. The second type (Coelho et al., 2017; Han et al., 2020; Iranmanesh et al., 2020b; Ren et al., 2022) reduces energy consumption of vehicles and drones by calculating the energy required to connect two locations (Han et al., 2020; Iranmanesh et al., 2020; Ren et al., 2022) or energy consumption at a specific time (Coelho et al., 2017; Iranmanesh et al., 2020; Ren et al., 2022). The third type (Liu, Deng, et al., 2021; Liu, Yan, et al., 2021) focuses on minimising the environmental impact of the delivery network by reducing the environmental effect of opening depots, satellite stations, delivering parcels, and operational handling.

Some studies focused on balancing cost and environmental impact with maximizing the demand coverage. However, these objectives often conflict with each other, as increasing the number of depots, drones, and vehicles to maximise demand coverage also increases costs and CO_2 emissions (Phan & Suzuki, 2016; Rashidzadeh et al., 2021). On the other hand, reducing these resources leads to lower costs and emissions but also reduced demand coverage (Chauhan et al., 2019; Rashidzadeh et al., 2021). Thus, studies aimed at maximising demand coverage with limited resources, such as a fixed number of depots, recharging stations, drones, and time windows, target the sustainability goals (Chauhan et al., 2019; Hong et al., 2018; Jung & Kim, 2022; Phan & Suzuki, 2016; Rashidzadeh et al., 2021).

1.6 MULTI-OBJECTIVE ARTICLES

This section highlights the presence of articles that simultaneously address two distinct goals, either minimising costs and environmental impact (Liu, Deng, et al., 2021; Liu, Yan, et al., 2021; Rashidzadeh et al., 2021; Resat, 2020) or minimising distance and facilities (Cai et al., 2021; Coelho et al., 2017; Phan & Suzuki, 2016; Pinto & Lagorio, 2022; Troudi et al., 2018). These articles fall into the category of multi-objective papers.

Multi-objective papers are divided into two types, with one type considering multiple objectives separately. For example, (Bakach et al., 2021) tackled the minimum number of hubs required by all scenarios by solving the first objective function, followed by the calculation of the optimal cost of vehicles and robots by solving the second and third objective functions.

(Dorling et al., 2017; Momeni et al., 2022; Peng et al., 2019) developed models and heuristics to address both minimising time and minimising cost objectives. However, there are articles that deal with conflicting objectives, such as maximising the satisfied demand and minimising the number of vehicles and travel distance (Phan & Suzuki, 2016). Liu et al. (2021) aimed to find the trade-off between the three conflicting goals of cost, environmental effect, and violation penalties. Reducing number of fleets to minimise costs may result in penalties for failing to meet customer demand on time, while using more vehicles to deliver parcels can minimise penalties but result in higher costs and emissions.

In the second category of multi-objective research, multiple objectives are taken into account in a single objective function. Unlike the non-dominant solutions proposed by (Liu et al., 2021), the scalarization technique introduced by Marler & Arora (2004) involves assigning a “weight” to each objective function, effectively transforming multiple objectives into a single scalar function to form a single-objective optimisation model (Pinto & Lagorio, 2022). In the function $min \sum_{w=1}^{W} \theta_w \cdot f_w(x)$, W is the number of objectives, θ_w represents the “weight” of the w-th objective function f_w. When the single objective function contains two objectives, it can also be written as $min\ \theta \cdot f_1 + (1 - \theta) \cdot f_2$ (Pinto & Lagorio, 2022) to minimise distance and the number of stations or $min\ \alpha \cdot f_1 + \beta \cdot f_2$ that $\alpha + \beta = 1$ (Cai et al., 2021) to minimise distance and the number of vehicles. In decision-making, it is crucial to assign specific values to each objective to prioritise their importance (Marler & Arora, 2004). For instance, Zhang & Yang (2022) placed a higher weight on customer satisfaction by assigning a 100 weight to the penalty cost for exceeding the customer service time and a 10 weight to the robot travel distance. Their focus was on optimising intelligent logistics to improve customer satisfaction. The objective function can also incorporate multiple objectives, as seen in the study by Han et al. (2020) who aimed to minimise the number of vehicles and energy consumption of both vehicles and drones. Troudi et al. (2018) aimed to minimise the total distance, number of drones, and batteries. Coelho et al. (2017) had even more objectives, with a focus on minimising the total distance, time, number, maximum speed, energy consumption of drones, and pickup and delivery parcel makespan - a total of seven objectives.

1.7 MAIN SOLUTION APPROACHES

According to Table 1.4, real-world logistics problems often encompass a substantial number of customers and an extensive number of parcels. To tackle these issues, various algorithms, and heuristics such as the Branch-and-Cut algorithm (referenced in 8 articles), Adaptive Large Neighbourhood Search (ALNS) (cited in 7 articles), Genetic Algorithm (GA) (mentioned in 7 articles), Simulated Annealing (SA) (referred to in 7 articles), and Local Search (LS) (discussed in 6 articles) are widely utilised to find solutions.

Table 1.4. Distribution of articles based on solution method.

Solution approach	Number**
Branch-and-Cut algorithm	8
Adaptive Large Neighborhood Search (ALNS)	7
Genetic Algorithm (GA)	7
Simulated Annealing (SA)	7
Local Search (LS)	6
Greedy Randomized Adaptive Search Procedure (GRASP)	4
Ant Colony Optimization (ACO)	3
Particle Swarm Optimization (PSO)	3
Tabu Search (TS)	3
Three-phased heuristic	3
Two-stage heuristic	3
Variable Neighborhood Search (VNS)	3
Branch-and-price algorithm	2
Clarke and Wright algorithm (CW)	2
Column Generation (CG)	2
Dynamic Programming	2
Immune Algorithm (IA)	2
Initiated Grouping Algorithm (IGA)	2
Minimum Marginal-Cost Algorithm (MMA)	2
Route, Transform, Shortest Path (RTS)	2
Truck and Drone Routing Algorithm (TDRA)	2
Others*	42

*: “Others” represents the subcategory with less than two articles.

**: The numbers indicate the number of articles involving that solution approach.

Out of the 8 articles on Branch-and-Cut algorithm, only one, (Cheng et al., 2020), has the objective of minimising cost. The rest of the articles focus on time minimization. In their study, Cheng et al. (2020) tackled the Drone Routing Problem (DRP) by considering a non-linear energy consumption function and associating each customer with a time window to minimise both travel and energy costs. They assumed that a fleet of drones would travel to multiple client locations, with each drone capable of making multiple stops on a single trip, but only able to be dispatched once from the depot. To solve their problem, Cheng et al. (2020) used a branch-and-cut algorithm to find the optimal routing and scheduling decisions for up to 50 customers. Dell'Amico et al. (2021a) also made contributions in this field by proposing three-indexed and two-indexed formulations and applying the branch-and-cut algorithm to minimise the completion time. In their research, they considered two waiting options for the drones – either at the customer locations or in the flying mode. The authors, (Dell'Amico et al. (2021b), further improved upon the related literature by developing four formulations, implemented as branch-and-cut algorithms, to minimise the arrival time at the depot. The study took into account multiple identical drones with battery restrictions and setup operations when multiple drones were launched or retrieved at the same node. They also evaluated the impact of having up to 5 drones on a single truck, serving up to 10 customers.

The objective of the Adaptive Large Neighbourhood Search (ALNS) articles varies, with only one of the seven articles aiming to minimise time, as opposed to the more commonly used Branch-and-Cut algorithm. Chen et al. (2021) proposed an ALNS heuristic to solve the Vehicle Routing Problem with Time Windows and Delivery Robots (VRPTWDR) with the goal of demonstrating the value of robots in urban areas and providing efficient solutions for up to 200 customers. Sacramento et al. (2019) used an ALNS metaheuristic to minimise cost in the capacitated multiple-truck problem with time window constraints. Jun Li et al. (2022) proposed an ALNS heuristic to solve the truck and drone routing problem with synchronisation and time window constraints for up to 100 nodes.

In the studies that use Genetic Algorithm (GA), Simulated Annealing (SA), or Local Search (LS), the objective function extends beyond minimising time or cost to include minimising fleet size (Al-Turjman et al., 2019), tardiness (Boysen, Schwerdfeger, et al., 2018 ; Jun et al., 2021), emission (Elsayed & Mohamed, 2020), flow (Hamzeei et al., 2013), or maximising covered demands (Hong et al., 2018). Due to the need for more advanced algorithms, hybrid heuristics and combined algorithms have emerged. For instance, Liu et al. (2020) applied a two-step Genetic Algorithm and Particle Swarm Optimization (GA-PSO) algorithm to improve delivery efficiency and minimise transport and emission costs with

AVs. Moshref-javadi et al. (2020) developed a mathematical formulation and a hybrid Tabu Search-Simulated Annealing algorithm to minimise customer waiting times in the multi-trip travelling repairman problem with drones. Ha et al. (2018) proposed two algorithms, TSP-LS and GRASP, to minimise operational costs, including transportation and waste time costs, in a study that extended the FSTSP with drones able to pick up and deliver items independently.

1.8 CHALLENGES POSED BY THE CURRENT LIMITATIONS

1.8.1 Lack of real data

Despite the selection of 103 articles for this review, only 24 of them (approx. one in five) contained actual case studies with real data, including studies conducted by (Chauhan et al., 2019; Coelho et al., 2017; Cokyasar et al., 2021; Dang et al., 2021; Elsayed & Mohamed, 2020; Ghelichi et al., 2021; Gu & Wallace, 2021; Hernández et al., 2020; Hong et al., 2018; Jeon et al., 2021), and others. The scarcity of real data in the industries with emerging logistics technologies leads to the use of simulated data sets in most research studies (Jeon et al., 2021; Nguyen et al., 2022; Z. Wang & Sheu, 2019). The lack of historical data or actual parameters makes it challenging to study the implementation of advanced technologies in the logistics industry, particularly in the area of last-mile logistics.

Collaboration with logistics companies can help overcome this challenge, as seen in case studies conducted by (Dang et al., 2021; Gu & Wallace, 2021; Scherr et al., 2020; S. Yu et al., 2022; G. Zhang et al., 2021). These partnerships can provide important information such as network layouts, demand scenarios, time frames, and fleet specifications. However, this is not always possible as logistics companies may be wary of sharing sensitive information.

Another strategy to gain insights into optimising vehicle routing for last-mile delivery is through the use of simulation models with auxiliary data. This includes simulations using geographical data for depot locations, as seen in studies conducted by (Bai et al., 2020; Campbell et al., 2018; Raj & Murray, 2020; Z. Wang & Sheu, 2019; J. J. Q. Yu, 2019; M. Zhang & Yang, 2022). Although these data sets may not be comprehensive, they can still provide valuable insights.

1.8.2 Current technological limitations and legal restrictions

The challenges facing emerging technologies in the logistics industry pose a threat to the growth and development of these advanced systems. The implementation of these technologies is dependent on smart infrastructure that can collect, share, and analyse data for efficient operations (Cheung et al., 2021). The lack of advanced technological infrastructure presents a major obstacle to the adoption of these technologies. Furthermore, the implementation of these technologies faces several challenges, including operability, security, realizability, and uncertainty in regards to government regulations (Elsayed & Mohamed, 2020; Rovira-Sugranes et al., 2022; Scherr et al., 2019). As an instance, the absence or ambiguity of rules and standards for drone operations, security requirements, privacy, and permissible airspace hinders the widespread use of drone technology (Rovira-Sugranes et al., 2022).

This is also a problem for autonomous vehicles and delivery robots. For example, the legality of autonomous vehicles and delivery robots on roads and liability for accidents is still uncertain (Scherr et al., 2019). Some states in the United States consider it legal for delivery robots to share the streets with people, with legal limits including a maximum speed of 12 mph on sidewalks and 25 mph on roadways, and a load limit of 550 pounds. Currently, delivery models mainly calculate Euclidean distance, ignoring actual road conditions and obstacles (H. Li et al., 2022; Jun Li et al., 2021; Moshref-Javadi et al., 2021; Poikonen & Golden, 2020; Puerto & Valverde, 2022; Resat, 2020; Xia et al., 2021). However, (Hong et al., 2018) took travel obstacles into account in their delivery network, resulting in a more accurate representation of the actual demand coverage and distribution network, despite a 20% decrease in demand coverage compared to Euclidean distance calculations.

1.8.3 Lack of uniform parameters and standards

The inconsistency in parameter values tested across various studies can lead to varying results. For instance, the flight speed of drones and the driving speed of vehicles can greatly impact the optimization of completion time and makespan, as seen in studies by (Dell’Amico et al., 2021a; Freitas & Penna, 2020; Gomez-Lagos et al., 2021; Raj & Murray, 2020; Schermer et al., 2020). Similarly, the unit travel costs or fixed costs for delivery robots, drones, and vehicles can have a significant impact on cost savings, as seen in studies by (Bakach et al., 2021; Coindreau et al., 2021; Cokyasar et al., 2021; Di Puglia Pugliese et al., 2021; Salama & Srinivas, 2020; Z. Wang & Sheu, 2019).

To address these issues, it is essential to establish uniform parameters and standards for testing and evaluating the efficiency and feasibility of delivery systems using drones, robots, and autonomous vehicles. This would require collaboration among researchers, industry experts, and regulatory bodies to develop a set of standardized parameters and guidelines for testing and evaluating delivery systems. Standardization would ensure that results are consistent and comparable across studies, enabling researchers to build upon each other's work and make meaningful advancements in the field.

Furthermore, the establishment of uniform parameters and standards would enable policymakers and regulators to make informed decisions about the use of these technologies in the delivery industry. With standardized testing protocols, policymakers would be better equipped to evaluate the potential benefits and risks of implementing drone and robotic delivery systems, leading to more effective and efficient regulation.

1.9 RESEARCH OBJECTIVES

The ongoing advancements in technology are continuously changing the world, transforming industry practices, and providing consumers with improved product and service options. Emerging technologies such as drones for remote distribution, robots, and autonomous vehicles for automatic distribution have increased the efficiency of the logistics process and service quality. While these technologies bring about opportunities for the logistics industry, they also present problems.

In recent years, optimizing the routes of drones and autonomous vehicles has become increasingly important as the demand for efficient and cost-effective last-mile delivery solutions grows. While previous studies have focused on mathematical models and optimization algorithms to address this problem, there is still a need to explore new models that can accommodate a variety of last-mile delivery platforms and real-world assumptions. Furthermore, it is crucial to consider the unique constraints and challenges associated with each industry and platform and develop customized optimization algorithms to effectively solve the problem. Therefore, the research question for this thesis is:

"How can we leverage innovative mathematical models and optimization algorithms to enhance the efficiency of last-mile delivery platforms in the context of emerging automated vehicle technologies?"

This thesis examines different platforms for vehicle routing optimization in last-mile pickup and delivery logistics with emerging technologies. The main topics of this study are as follow:

Development of Route Optimization Mathematical Models and Algorithms

One significant area of research is the development of mathematical models and algorithms that can optimize the delivery routes of autonomous vehicles and drones for last-mile delivery operations. These algorithms take into account factors such as delivery location, package size, customer constraints, synchronization among vehicles to determine the most efficient delivery method for each last-mile delivery platforms.

Multi-Modal or Hybrid Delivery Solutions

Autonomous vehicles and drones provide a variety of options for last-mile delivery. One main research direction is to identify the most efficient and effective mode of transportation for a given delivery task, which can improve the overall delivery efficiency.

1.10 RESEARCH GAP

The optimization of pickup and delivery operations in a two-echelon network with simultaneous pickup and delivery operations has been a crucial area of focus within the logistics domain. However, existing literature and methodologies have exhibited certain limitations in addressing the complex dynamics inherent in such operations. This research identifies and addresses these gaps through the following significant contributions:

1.10.1 Development of a Comprehensive Framework of Autonomous Mobile Lockers- Courier Delivery Operations

The proposed framework encompasses an Autonomous Mobile Locker (AML)-Courier routing problem, a scenario previously rarely covered in existing models. An Autonomous Mobile Locker (AML) is a type of self-contained, mobile storage unit equipped with autonomous navigation capabilities. These lockers are designed to move independently from one location to another, often in urban or semi-urban environments, without human intervention. They are envisaged for various logistics and delivery operations to optimize the storage and transportation of goods (Jun Li et al., 2021). This innovation enables couriers to execute multiple tours in the second echelon, enhancing operational efficiency. Furthermore, the allowance of open tours in both echelons expands the potential for couriers to serve customers across various residential zones. The formulation of a novel cost optimizing Mixed

Integer Linear Programming (MILP) model (2E-MOVRP) specifically tailored to address the complexities of the 2-echelon pickup and delivery problem stands as a pioneering achievement in this field. The development of an adaptive matheuristic further fortifies the framework's applicability to large-scale networks.

1.10.2 Establishment of a Flexible Cooperation System of Drones and Trucks

Introducing a cooperative system between drones and trucks represents a significant advancement in the realm of delivery operations. The *m*-drone, *n*-truck routing framework, designed to allow drones to be retrieved by any nearby truck, offers a level of flexibility previously absent from conventional systems. Incorporating various drone dispatch and collection strategies, including dispatch-wait-collect and dispatch-move-collect, enriches the operational repertoire of this cooperative system.

1.10.3 Generalization of Drone-Truck Routing with Multiple Time Windows

This research extends the scope of the drone-truck routing problem by considering multiple time windows for each customer. Allowing customers to choose one or multiple time windows provides a more adaptive approach to scheduling. The formulation of an exact Constraint Programming model for determining optimum scheduling decisions complements the existing literature in this domain. Additionally, the introduction of a novel cost optimizing MILP model addresses this generalized problem, while the Adaptive Large Neighbourhood Search (ALNS) heuristic approach enhances the computational efficiency of the solution, particularly for large-scale instances.

These contributions collectively represent a significant stride forward in the field of delivery operations. By providing more comprehensive and flexible solutions, this research bridges critical gaps in the existing literature and establishes a new standard for operational excellence in logistics.

1.11 THESIS OUTLINE

In Chapters 2, 3, and 4, we delve into the development and optimization of innovative delivery systems designed to revolutionize the logistics landscape. Each of these chapters addresses a critical aspect of modern delivery operations, building upon the principles of efficiency, flexibility, and cost-effectiveness.

Chapter 2 introduces a hybrid delivery platform featuring Autonomous Mobile Lockers (AMLs) and couriers, setting the stage for our exploration of simultaneous pickup and delivery route planning within a two-echelon parcel distribution network. This chapter lays the foundation for the subsequent discussions by presenting a mathematical model and solution approach tailored to this complex logistical challenge.

Chapter 3 shifts our focus to the integration of drones and trucks in a flexible Truck-Drone Delivery System. We introduce the concept of leveraging drones to augment truck deliveries, emphasizing the optimization of interconnected routes to minimize logistics costs. Our proposed model for route planning provides a framework for achieving this goal.

Chapter 4 tackles the intricacies of delivering e-groceries, highlighting the limitations of drone deliveries for heavy and bulky items. To address these constraints, we unveil a collaborative truck-drone delivery system that offers customers multiple time windows for deliveries, enhancing efficiency and flexibility. The chapter also introduces a Mixed Integer Programming (MIP) model and a hybrid Adaptive Large Neighbourhood Search (ALNS) and Constraint Programming (CP) model, which play pivotal roles in optimizing the delivery system.

Chapter 5 provides a summary of the contributions and main findings of each chapter. The implications of the proposed collaborative delivery systems for the future of the delivery industry are also discussed. Finally, areas for future research and potential improvements to the proposed systems are highlighted.

These chapters explore the adaptability and innovation in delivery operations offered by the intersection of autonomous mobile lockers, drones, trucks, and advanced optimization techniques to form a comprehensive framework with the potential to reshape the future of delivery services. The following chapter provides a comprehensive summary of our contributions and their implications for the delivery industry, paving the way for future research and enhancements in these ground-breaking systems.

Chapter 2: Cost-optimal Deployment of Autonomous Mobile Lockers Co-operating with Couriers for Simultaneous Pickup and Delivery Operations

2.1 INTRODUCTION

Due to the nature of online shopping, customers generally tend to receive their purchased goods at their doorstep. Therefore, one of the results of the general popularity of online shopping is the increase in the volume of transportation of goods from stores or distribution centres to customers' residences (Bergdoll & Christensen, 2008). This itself increases the volume of urban traffic, given that a high percentage of this transportation is performed by traditional ground vehicles such as vans or trucks. The use of gasoline-powered trucks, in addition to causing road congestion, also gives rise to environmental problems and increases the cost of transporting goods.

Advances in electrification and artificial intelligence have brought about fundamental changes in the logistics and transportation industries. With emerging advanced autonomous vehicles including drones, delivery robots, self-driving cars, and driverless trucks, the parcel logistics industry has new opportunities in recent years (Chen et al., 2021). Three categories of last-mile delivery robots have been described in the literature: Wheeled Sidewalk Pods (Droids), Drones, and Autonomous Ground Vehicles (AGVs) with lockers or Autonomous Mobile Lockers (AMLs) (Joerss et al., 2016). Figure 2.1 shows extant examples of the three types of last-mile delivery robots.

AMLs have significant advantages for large-scale last-mile urban parcel delivery because they can carry large payloads, are immune to adverse weather events, are environmentally friendly and have low operating costs (Jun Li et al., 2021). AMLs have the potential to be a sustainable solution for logistics companies.

In current dense urban parcel delivery systems (such as in Shenzhen, China) walking couriers (or couriers, for brevity) perform direct tours, picking up parcels at a depot and delivering them to end customers. Parcel companies value direct contact with customers. Moreover, couriers can collect proof of delivery and make decisions about whether and where to leave parcels when the customer is not there. Due to couriers' speed and carrying capacity, depots can only service relatively small areas, increasing the depot number increases operational cost for the parcel company.

Droid

https://www.wired.com/story/amazon-new-delivery-robot-scout/

UAV

https://uavcoach.com/amazon-drones-trucks/

AML

https://www.scmp.com/tech/tech-trends/article/3134898/meituan-jdcom-li-auto-backed-neolix-win-licences-test-unmanned

Figure 2.1. Different delivery robots.

In order to improve the efficiency of parcel delivery systems in dense urban areas Jun Li et al. (2021) proposed a simultaneous pickup and delivery routing platform comprising a cooperative system of AMLs and couriers in a two-echelon network. In the first echelon, AMLs transfer parcels to and from stops where they meet the couriers. In the second echelon, the couriers pick up and deliver the parcels as they visit customers and transfer the parcels to the AMLs at the stops. The model presented by Jun Li et al. (2021) considers the fixed cost of hiring couriers to deliver and collect parcels. Couriers were only allowed to implement one route, meaning that couriers could not be sent on another mission after completing a route, and the system was imposed the same fixed cost for even a small route as for an extensive route. The current study improves on the model presented by Jun Li et al. (2021) by allowing couriers to perform multiple tours, which may lead to a reduction in the number of couriers used and, subsequently, the fixed costs.

In addition, Jun Li et al. (2021) programmed a pickup and delivery model for an integrated regional postal service where asset monopoly does not seem necessary. AMLs and couriers are both allowed to perform open routes, which results in a more flexible routing platform. The open vehicle routing problem (OVRP) allows a vehicle to end its mission at any other depots which is not necessary the same depot from which it started its tour. By doing so, AMLs are no longer restricted to servicing a single depot and a courier may meet an AML at different stops. For reasons of consistency, we assume conservation of AMLs at depots, although the specific AMLs at a particular depot at the end of the period may be different from those at the start. The period under consideration for the AMLs and couriers is left open but would be a fraction of a day (somewhere between one to eight hours).

Figure 2.2 illustrates this concept with two AMLs, nine stops, six courier tours, and four couriers. This chapter refers to this option as the "multi-trip" or "multiple trip" approach, in alignment with existing literature on the subject. Note that in the second echelon, couriers have the capability to provide pickup and delivery services to multiple customers during a single tour.

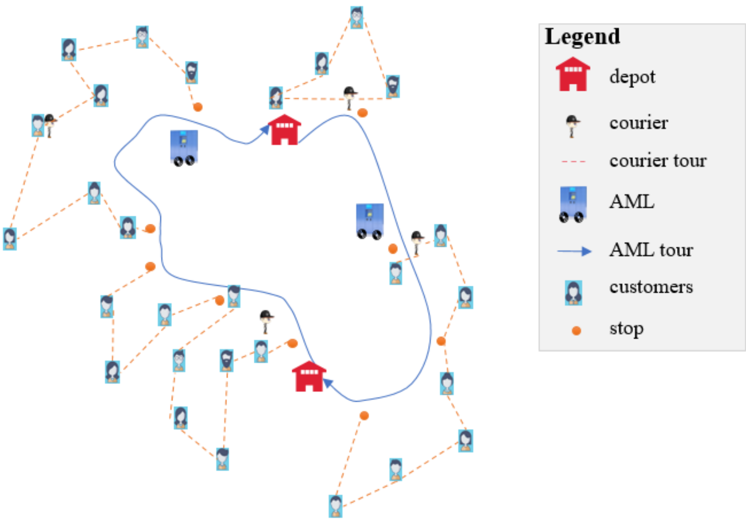


Figure 2.2. Two-echelon multi-depot multi-trip open vehicle routing problem

2.1.1 Contributions

This study is an extension of previous work by Jun Li et al. (2021) and utilizes a similar network. Moreover, this is believed to be the first study that allows ‘open’ routes (routes that end at another depot or stop) and multiple trips in a two-echelon network with simultaneous pickup and delivery operations. Also, at both echelons, couriers and AMLs can perform open routes if it leads to a reduction in cost. Note that performing open routes is not imposed by the model. Rather, open routes happen when it leads to a cost saving.

The main contributions of this chapter are as follows:

- Enhanced Courier Flexibility: This research introduces a framework that empowers couriers to conduct multiple tours within the second echelon, significantly augmenting their operational capacity.
- Residential Zone Flexibility: The proposed model allows for open tours in both echelons, enabling couriers to efficiently serve customers in diverse residential zones, enhancing accessibility and service coverage.
- Innovative Optimization Model: A novel Mixed Integer Linear Programming (MILP) model, is developed to tackle the 2-echelon pickup and delivery challenge, accommodating multiple courier tours and open routes. This optimization model stands as a pioneering advancement in this domain.
- Comparative Analysis for Best Practice: The proposed model is rigorously compared with other variants of simultaneous pickup and delivery problems, including the work by Jun Li et al. (2021), to establish the most effective approach and identify best practices in this context.

The remainder of this chapter is organized as follows. In Section 2.2 a review of related literature is presented. Section 2.3 defines the research problem, and the mathematical formulation is presented. Section 2.4 proposes a solution methodology. In Sections 2.5 and 2.6, results and sensitivity analyses for a range of problem sizes are presented and discussed, respectively. Finally, the conclusion is provided in Section 2.7.

2.2 RELEVANT PREVIOUS WORKS

This section reviews the related literature and provides a comparative analysis with previous work. We survey similar studies in which robots are deployed to perform last-mile delivery operations. Then, we identify the research gap and present an overview of the main differences between our work and the existing studies. Table 2.1 shows the differences between this work and the most related studies in the literature.

2.2.1 Vehicle Routing Problem with Drones and Droids

The deployment of delivery robots has attracted the attention of many researchers in recent years. The introduction of a cooperative system, in which robots typically assist ground vehicles in delivery operations, increases the efficiency of the transportation system. In the relevant literature, delivery robots are divided into two general categories: UAVs, otherwise known as drones, and small ground-based robots, sometimes referred to as droids.

Due to the differences in the technical characteristics of drones and ground robots, studies on drone delivery systems are also different in terms of model structure. The most obvious difference is in the rules regarding the flight of drones in urban areas. In many countries, drones are banned from flying over densely populated urban areas due to safety concerns. Conversely, small delivery robots moving on sidewalks at low speed can meet customers living or working in areas where heavy ground vehicles are generally prohibited from operation or find difficulty parking.

The hybrid vehicle-drone delivery routing problem was first introduced by Murray & Chu (2015) and many variants of this problem have since been studied by academics. In its basic form, known as the 'Flying Sidekick Traveling Salesman Problem' (FSTSP), a truck equipped with a drone performs delivery operations for a set of customers, with the truck acting as a mobile depot. A more flexible variant of FSTSP, known as the Traveling Salesman Problem with Drone (TSP-D), was first introduced by Agatz et al. (2018). The difference between TSPD and FSTP is in the drone's ability to perform multiple deliveries per mission. Subsequent studies challenged some assumptions existing in FSTP and attempted to remove the simplifications. For example, studies typically known as Vehicle Routing Problem with Drones (VRP-D) are mostly conducted with the aim of minimizing cost or makespan when a truck can carry more than one drone. For example, (Kitjacharoenchai, Min, et al., 2019) conducted a study in which a truck equipped with a maximum of two drones delivers purchased items to end customers with the aim of minimizing the total time required to complete all deliveries. Wang & Sheu (2019) proposed a model in which the truck does not have to wait at the docking stops to collect the drones. Rather, trucks can continue their journey and collect the drones at subsequent stops.

Another variant of the vehicle-drone collaborative delivery system, known as the Multi-visit Vehicle-Drone Routing Problem (MVRP-D), is the most sophisticated form of the VRP-D so far. The MVRP-D, in its most advanced form, consists of multiple trucks, which in cooperation with multiple drones, deliver shipments to end customers. In the MVRP-D, also, it is assumed that the drones have enough payload capacity to visit more than one customer per mission. Taking into account the energy consumption of the drone, (Luo et al., 2021) formulate a mixed-integer linear program (MILP) model addressing the multi-visit traveling salesman problem with multi-drones (MTSP-MD). The objective of this study is to minimize the time required by the truck and the drones to implement the deliveries. They also propose a multi-start tabu search (MSTS) algorithm with tailored neighbourhood structure and a two-level solution evaluation method to solve the problem for real-size instances.

In the drone-truck routing problem, a drone can be retrieved at the same site from where they were dispatched or at the next truck stop. Each drone may also travel multiple times. Our problem and the truck-drone routing problem are quite similar due to these two characteristics. However, our study differs from the UAV-truck routeing problem (VRP-D) in a few key ways. In contrast to the AML-courier routeing problem, where AMLs and couriers independently travel their routes, drones and trucks move in tandem in the VRP-D. Due to security and safety considerations, synchronisation between trucks and drones is necessary in VRP-D. Unlike on drone-truck routing problems, where drones must hover above the rendezvous sites until picked up by a truck, both AMLs and couriers are not in danger if they wait at the rendezvous points.

However, the synchronization between drones and trucks is frequently overlooked in studies (Dayarian et al., 2020; Ferrandez et al., 2016; Poikonen & Golden, 2020; Savuran & Karakaya, 2016; Ulmer & Thomas, 2018; Wang & Sheu, 2019; Kang & Lee, 2021). For instance, Kang & Lee (2021) proposed an exact algorithm for the drone-truck routing problem without taking into account the synchronization between drones and trucks in their newly published study in Transportation Science. According to them, the inclusion of the synchronization requirements for the drone-truck coordination makes the problem considerably more complicated to solve, necessitating the use of more general heuristics and metaheuristics that provide solutions with less accuracy. The vehicle routing problem with drones was addressed by Wang & Sheu (2019), who also presented a branch-and-price method in which they did not take into account truck and drone synchronization. In this study, the synchronization between AMLs and couriers is not considered as waiting does not pose a security problem, thereby avoiding unnecessary computational complexity.

This study primarily analyses the cost of the AML-courier routing problem. However, two parameters control the maximum courier and AML range to guarantee that all deliveries can be accomplished within a single day. The total daily distance that a courier and AML can travel can be set using these two parameters. Given that couriers can complete multiple journeys, we can handle the waiting time at the rendezvous if we choose small values for these parameters. For example, if the maximum range of the AML is such that the tour duration is limited to half a day, this would allow the couriers a minimum of half a day to complete their pickup and delivery operations within the day. We could then choose courier range limits to ensure this is feasible. Additionally, if the maximum range of an AML is low, AMLs will travel shorter distances and courier visits will occur more frequently, reducing the amount of time spent waiting for couriers. In addition, the model allows for both closed and open routes, so if an open route is impractical or cost-inefficient with respect to the range limits, a closed route will be taken.

As mentioned, drones, although much more manoeuvrable than ground robots, are commonly used to deliver parcels in sparsely populated and remote areas. Especially drones that can perform multi-visit routes, due to their large size, require a relatively large space for landing, which is not feasible in densely urban areas. As a practical solution, small robots can fill this operational gap. These small robots, whose payload capacity does not exceed 10-15 kg, can be carried by a van and deliver parcels to customers from the van. (Boysen, Schwerdfeger, et al., 2018) studied the scheduling of truck-based autonomous robots for a delivery problem, formulated as a MIP model. In this study, a truck loaded with a number of delivery robots cooperatively performs the home delivery operation. The trucks function as mobile depots from which the robots are dispatched to customers, while the truck itself is also responsible for delivery. They concluded that the hybrid robot-van delivery system involves a lower cost compared to the traditional system in which a truck only is deployed. Finally, they proposed a heuristic whereby they could solve the problem for large instances.

Table 2.1. Selected studies on the most relevant literature

Study	Vehicle Type	Pickup/Delivery	Multiple Depot	Multiple Vehicle	2-Echelon/ 1-Echelon	Multi Trip	Open Routes	Multi/Single Visit	Approach	Objective Function	Solution
(Murray & Chu, 2015)	Drone/Truck	D	-	-	2E-VRP	-	-	S	MIP	Makespan	Heuristic
(Agatz et al., 2018)	Drone/Truck	D	-	*	2E-VRP	*	-	M	MIP	Makespan	Heuristic
(Kitjacharoenchai, Min, et al., 2019)	Drone/Truck	D	*	*	1E-VRP	*	-	M	MIP	Makespan	Heuristic
(Luo et al., 2021)	Drone/Truck	D	-	*	1E-VRP	*	-	M	MIP	Makespan	Heuristic

(Simoni et al., 2020)	Robot/Van	D	-	-	2E-VRP	-	-	M	MIP	Cost	Heuristic
(Boysen, Briskorn, et al., 2018)	Robot/Van	D	-	-	2E-VRP	-	-	S	MIP	Cost	Heuristic
(Chen et al., 2021)	Robot/Van	D	-	-	2E-VRP	-	-	S	MIP	Cost	Heuristic
(Yu et al., 2020)	Robot/Van	D	-	*	2E-VRP	-	-	M	MIP	Cost	Heuristic
(Alfandari et al., 2021)	Robot/Van	D	-	-	2E-VRP	-	-	S	MIP	Tardiness	Exact
(Ostermeier et al., 2021)	Robot/Truck	D	-	-	2E-VRP	-	-	S	MIP	Cost	Heuristic
(Jun Li et al., 2021)	AML-courier	PD	*	*	2E-VRP	-	-	M	MIP	Cost	Heuristic
This study	AML-courier	PD	*	*	2E-VRP	*	*	M	MIP	Cost	Heuristic

D: Delivery; P: Pickup; PD: Simultaneous Pickup and Delivery; 2E-VRP: 2 Echelon Vehicle Routing Problem; M: Multi Visit; S: Single Visit, MIP: Mixed Integer Programming

Simoni et al. (2020) also investigated a collaborative delivery problem consisting of a truck loaded with a robot and modelled it using MILP. Then, they introduced a heuristic method to solve the problem of up to 100 customers. By conducting sensitivity analysis on important parameters of the problem, they concluded that the time saving resulting from the use of the robot-truck system is highly dependent on the speed ratio between truck and robot as well as the carrying capacity of the robot. Yu et al. (2020) also addressed the same problem where multiple robots can be carried by a truck, each of which can perform multiple visits within their routes. They formulated their problem as a MILP model whose objective function minimizes the total transportation cost and developed a heuristic and a hybrid metaheuristic to deal with large networks. They concluded that the cost saved by increasing the robots' speed is marginal and suggested using low-speed robots to be more pedestrian-friendly. However, the delivery robots cannot be deployed multiple times for different missions, which may lead to using a higher number of robots. Also, their network consists of a single depot and open routes are not allowed either in the first echelon or in the second echelon.

Chen et al. (2021) developed an Adaptive Large Neighbourhood Search (ALNS) heuristic algorithm to solve a problem in which a truck equipped with several self-driving delivery robots distribute parcels to end customers in a two-echelon network. The objective of this study is to find the near-optimum routing and scheduling decisions when delivery time windows are taken into account. Alfandari et al. (2021) addressed the robot-vehicle parcel delivery problem where multiple robots depart the vehicle to serve customers. They devised a MILP formulation to minimize a tardiness indicator with respect to the customer deadlines, and a tailored Benders decomposition approach to solve realistically sized instances. Yu et al. (2021) formulated the robot-vehicle pickup and delivery problem as a MILP model whose objective function minimizes total cost. They solved their problem by developing an ALNS heuristic algorithm for large networks. They also examined the impact of the robot travel cost and maximum travel range on the total cost. Ostermeier et al. (2022) addressed a truck-robot delivery routing problem with time windows, formulated as a MILP model whose objective function seeks to minimize total cost. For large instances, they developed a heuristic and concluded that the truck-robot delivery system can reduce last-mile delivery cost by up to 68% compared to the truck-only system.

2.2.2 Multi-trips Vehicle Routing Problem (MVRP)

The multi-trip vehicle routing problem (MTVRP) assumes that a vehicle can execute more than one route per day without exceeding its maximum daily travel time, which was first discussed by Fleischmann (1990). Many algorithms not only solve the basic MTVRP but also solve its variants. For example, François et al. (2016) proposed large neighbourhood search (LNS) heuristics for the traditional MTVRP, which only considered the delivery routes. Wassan et al. (2017) developed a two-level variable neighbourhood search (VNS) algorithm for the MTVRP with backhauls (MT-VRPB) for the one echelon problem. Grangier et al. (2016) solved the two-echelon multi-trip vehicle routing problem (2E-MTVRP) for delivery with an adaptive large neighbourhood search (ALNS) algorithm. However, they did not consider the simultaneous pickup and delivery.

Zhou et al. (2018) introduced a hybrid multi-population genetic (HMPG) metaheuristic for the two-echelon multi-depot vehicle routeing problem (2E- MDVRP). Zhen et al. (2020) studied a hybrid particle swarm optimisation algorithm (HPSO) and a hybrid GA to solve the multi-depot multi-trip VRP with time windows considering release dates (Multi-D&T VRPTW-R), assuming that each trip is constrained to begin and end at the same depot.

According to the recently published review paper by Sluijk et al. (2022) on the two-echelon vehicle routing problem, one potential research direction is the 2E-VRP with multiple trips on the second echelon. In our model, couriers on the second echelon are allowed to perform multi-trip missions. Although a lower carrying capacity in fact equates to a shorter tour length, in this study the key justification for considering multiple courier trips is the assumed lower carrying capacity of couriers compared to AMLs.

2.2.3 Open Vehicle Routing Problem (OVRP)

On this foundation, Nucamendi-Guillén et al. (2021) adopted the notion of open routeing, in contrast to the numerous multi-depot models where the vehicle travels on closed routes and returns to the same departing depot. The open vehicle routing problem (OVRP), where the trucks do not need to return to the original depot after completing deliveries to customers, was first discussed by Schrage (1981) and has been investigated by researchers and industry since 2000 (F. Li et al., 2007; C. Lin et al., 2014; Sariklis & Powell, 2000).

Recent studies have considered the multi-depot open vehicle routeing problem, in contrast to earlier studies, which only addressed the OVRP with a single depot (MDOVRP). For instance, Brandão (2020) introduced a memory-based iterated local search algorithm (MBILSA) to solve MDOVRP, while Soto et al. (2017) developed a multiple neighbourhood search hybridised with a tabu search (MNS-TS) approach. However, even though multiple depots in their studies allowed each route to begin at any depot, the routes terminated at the last visited customer by disregarding the last return trip to other depots. By contrast, we allow AMLs and couriers to start and end at any depots or stops, respectively. Yu et al. (2020) proposed a MIP model and a hybrid metaheuristic to solve the two-echelon OVRP. While they only explored the 2nd-echelon open routes with a single depot, we consider the open routes of AMLs and couriers for both echelons with multiple depots and multiple stops.

To the best of our knowledge, only An (2022) addressed closed-open routing, multi-trips, and differing vehicle kinds simultaneously. Our study is more extensive since we additionally take into account multiple depots, simultaneous pickup and delivery, and open routing in both echelons.

2.2.4 Trailer-truck Routeing Problem (TTRP)

Due to congested roadways, and other practical limitations, the trailer must stop at a parking spot and wait for the truck to be picked up in the truck and trailer routing problem (TTRP). An AML, by contrast, does not remain stationary like a trailer. AMLs can move to another stop to exchange packages with couriers, or it can return to the same or a different depot. AMLs and couriers independently traverse their routes, in contrast to trailers that must be moved by trucks (Bartolini & Schneider, 2020).

This study is an extension to the work carried out by Jun Li et al. (2021) in which a hybrid system of AML-courier performs pickup and delivery tasks in a two-echelon network. To solve their problem for large network instances, they extended the Clarke and Wright algorithm. Unlike the model developed by Jun Li et al. (2021), our study assumes that couriers can perform multiple missions while in both echelons open routes are allowed. This assumption is consistent with real-world practice where multiple routes can be assigned to one courier. This reduces the fixed cost associated with employing couriers. In addition, this study differs from the study done by Jun Li et al. (2021) in terms of the solution methodology.

Analogous to the studies reviewed, this chapter addresses the deployment of autonomous delivery robots in a two-tier network. However, this study considers the autonomous mobile lockers (AML) in the first echelon connecting depots to couriers in the second echelon. The AMLs allow couriers to commence a new tour after completing the previous tour. Moreover, in both echelons, the AMLs and couriers can perform open tours, which increases system flexibility and cost savings.

This chapter considers m-AMLs and n-couriers in a parcel pickup and delivery network consisting of multiple depots and several stops. AMLs move between depots and the stops while couriers deliver to customer locations. We formulated our problem as a MILP model whose objective function aims at minimizing the total fixed and variable costs. Furthermore, we develop an adaptive matheuristic approach to handle large-size instances.

2.3 PROBLEM DEFINITION

The problem under investigation addresses simultaneous pickup and delivery route planning in a two-echelon parcel distribution network, where $G(N, E)$ represents a network consisting of N nodes and E links. A hybrid robot-human system serves end-customers in which robots, operating as AMLs, meet the couriers at 'stops' (public places accessible to pedestrians and bikes without any special infrastructure) to exchange parcels. The set of network nodes comprises multiple depots $N_0 \subset N$, AML stops $N_V \subset N$, and customer points $N_C \subset N$.

The nodes to which AMLs can travel are $(N_0 \cup N_V)$, while couriers can move between customer locations and stops $(N_C \cup N_V)$. A set of couriers represented by C, in cooperation with a set of AMLs denoted by A deliver parcels from depots to customers and transfer their pickups to depots. AMLs transport parcels from depots to stops, from where couriers begin their tours to customer locations. AMLs also convey pickups collected by couriers to depots. Couriers can perform multiple missions, meaning that couriers can complete a series of missions each of which begins and ends at stops (not necessarily the same stop as open routes are permitted). Note that a courier only travels to an AML stop when there is at least one parcel to be delivered to a customer or at least one parcel to be transferred to a depot. In addition, open routes are allowed for AMLs, meaning that AML missions do not need to begin and end at the same depot, subject to the number of AMLs at each depot being the same at the beginning of the period of analysis as at the end.

The decision to exclude time windows from this research is primarily motivated by the need for simplification and practical relevance. The problem at hand already involves a high degree of complexity, featuring a two-echelon simultaneous pickup and delivery system, open routes, and the capacity for multi-trip operations. Given this inherent complexity, the inclusion of strict time windows would further escalate the intricacy of the model, potentially making it unwieldy and less practical for real-world logistics applications.

Furthermore, the practical context of the problem must be acknowledged. In many logistics scenarios, particularly those involving the use of lockers or mailboxes, the imposition of stringent time windows may not be necessary. Parcels placed in such secure storage locations are typically immune to issues of deterioration or time sensitivity, making the consideration of time windows less critical for these specific delivery purposes.

By omitting time windows, this research can focus on addressing the core logistical challenges presented by the integration of AMLs, couriers, open routes, and multi-trip operations. This approach allows for the development of a more practical, manageable, and relevant solution that aligns with the specific problem domain and objectives of this study. The aim of the solution process is to find the routing decisions such that the total logistics cost is minimized.

The assumptions considered in this study are as follows:

- Limited Capacity of AMLs and Couriers: This assumption mirrors real-world constraints where delivery vehicles, including AMLs and couriers, have finite carrying capacities. These limitations are essential to maintain operational efficiency, load balance, and adherence to safety regulations.
- Predetermined Range for Distance Travel: The limitation on the total distance travelled is grounded in practical considerations. It helps manage fuel costs,

driver fatigue, and delivery timeframes, ensuring that the delivery process remains cost-effective and timely.

- Simultaneous Pickup and Delivery Demand: Recognizing that each customer can have both pickup and delivery demands concurrently aligns with common scenarios where items need to be retrieved from one location and delivered to another for customers. This simplifies modelling real-world scenarios.
- AMLs Restricted from Traveling to Customer Locations: Restricting AMLs from traveling to customer locations is often based on technological and safety considerations. AMLs might not be designed for navigating complex customer environments and are better suited for accessing fixed pickup/drop-off points.
- Couriers Performing Multiple Missions: Allowing couriers to undertake multiple missions within their range is an efficiency-driven assumption. It acknowledges that couriers, often human agents, can handle multiple deliveries during a single route, optimizing resource utilization.
- Open Routes for AMLs and Couriers: Open routes recognize the adaptability of delivery systems. Allowing AMLs and couriers to perform open routes accommodates unforeseen customer demands or shifts in delivery priorities, enhancing flexibility.
- AMLs Completing One Tour per Day for Next-Day Delivery: This assumption aligns with the concept of next-day delivery, which is a common practice in the logistics industry. It simplifies the operational timeline and planning, ensuring that customers receive their orders promptly while also managing operational complexity.

2.4 PROBLEM FORMULATION

This section presents a mathematical formulation and the notations used. Since this study is an extension of the work presented by Jun Li et al. (2021), the notations are similar for ease of tracing. Variables x_{ijt}, h_{it}, and y_t determines the route sequences. Variables z_{ij} and f_{ij} determine the delivery and pickup assignments for each echelon. In order to control the capacity constraints and monitor the parcels flow at each stop, variables q_{it}, u_{ij}, and v_{ij}

are defined. Variables s_{it}, e_{it}, b_{it}, w_{ji}^1, and w_{ji}^2 cooperatively function to build multi and single trips in the second echelon.

Sets

N: The set of all nodes;

N_0: The set of depots, $N_0 \subset N$;

N_C: The set of customer locations, $N_C \subset N$;

N_V: The set of stops at which AMLs and couriers exchange parcels, $N_V \subset N$;

A: The set of AMLs;

C: The set of couriers;

T: The set of AMLs and couriers, $T = C \cup A$;

$\mathcal{F}$: The set of second echelon arcs, visited by couriers between stops and customers ($\mathcal{F} = \{(i,j): \forall i \in N/N_0,\ \forall j \in N/N_0, i \neq j\}$);

$\mathcal{L}$: The set of first echelon arcs, visited by AMLs between depots and stops. $\mathcal{L} = \{(i,j): \forall i \in N/N_C,\ \forall j \in N/N_C, i \neq j\}$.

Parameters

c_{ijt}: Unit traveling cost of an AML or a courier $t \in T$ per unit distance from node $i \in N$ to node $j \in N$;

$dist_{ij}$: Distance from node $i \in N$ to node $j \in N$;

d_i: Delivery demand at node $i \in N_C$;

p_i: Pickup demand at node $i \in N_C$;

CV: The capacity of each AML;

CC: The capacity of each courier;

CD_i: The capacity of depot $i \in N_0$;

M: A sufficiently large positive number;

rv: Maximum travel range of an AML;

rc: Maximum travel range of a courier;

FV: AML fixed cost per day;

FC: Courier fixed cost per day.

Decision variables

x_{ijt}: A binary variable which takes 1 if an AML or a courier $t \in T$ travels from node $i \in N$ to node $j \in N$ where $i \neq j$, and 0 otherwise;

y_t: A binary variable which takes 1 if an AML or a courier $t \in T$ is employed, and 0 otherwise;

z_{ij}: A binary variable which takes 1 if delivery demand of costumer $i \in N_C$ is assigned to stop $j \in N_V$, and 0 otherwise;

f_{ij}: A binary variable which takes 1 if pickup demand of costumer $i \in N_C$ is assigned to stop $j \in N_V$, and 0 otherwise;

q_{it}: A binary variable which takes 1 if stop $i \in N_V$ is visited by AML $t \in A$, and 0 otherwise;

u_{ij}: The residual weight of parcels for delivery before node $j \in N$, traveling from node $i \in N$;

v_{ij}: The cumulative weight of parcels picked up before node $j \in N$, traveling from node $i \in N$;

h_{it}: The order of node $i \in N_V$ in the route of AML $t \in A$;

s_{it}: A binary variable which 1 if courier $t \in C$ starts its mission from stop $i \in N_V$, and 0 otherwise;

e_{it}: A binary variable which takes 1 if courier $t \in C$ ends its mission at stop $i \in N_V$, and 0 otherwise;

b_{it}: A binary variable which takes 1 if courier $t \in C$ performs at least one open mission from stop $i \in N_V$, and 0 otherwise;

w_{ji}^1: A binary variable which takes 1 if only the pickup task of costumer $i \in N_C$ is assigned to stop $j \in N_V$, and 0 otherwise;

w_{ji}^2: A binary variable which takes 1 if only the delivery task of costumer $i \in N_C$ is assigned to stop $j \in N_V$, and 0 otherwise.

Considering an objective function that minimizes the total logistics costs, the problem is modelled in the form of an MIP as presented below:

$$min \sum_{i \in N, j \in N, t \in T} x_{ijt} c_{ijt} + \sum_{i \in N, j \in N, t \in A} FV y_t + \sum_{i \in N, j \in N, t \in C} FC y_t \quad (1)$$

Equation (1) minimises the total logistics cost, comprising three cost components, namely the operational cost followed by the fixed cost for the AMLs and finally by the fixed cost for the couriers. Note that operational cost varies with respect to the operations. Pick-ups and deliveries carried out by couriers yield higher operational costs compared with AMLs whose function is to serve the couriers. The fixed costs are calculated based on the daily rates of employing AMLs and couriers.

The first echelon constraints:

$$\sum_{j \in N, t \in A, (j,i,t) \in \mathcal{L}} x_{jit} \leq \sum_{k \in N_C} z_{ki} \qquad \forall i \in N_V \qquad (2)$$

$$\sum_{j \in N, j \notin N_C, i \neq j} x_{jit} = \sum_{j \in N, j \notin N_C, i \neq j} x_{ijt} \qquad \forall i \in N_V, t \in A \qquad (3)$$

$$\sum_{j \in N_V, t \in A} x_{jit} = \sum_{j \in N_V, t \in A} x_{ijt} \qquad \forall i \in N_0 \qquad (4)$$

Constraint (2) enforces the AMLs to visit the stops to which a set of customers to be served by couriers are assigned. Constraints (3) and (4) ensure AML conservation in the first echelon.

$$\sum_{i \in N_0, j \in N_V, j \notin N_0} x_{ijt} \leq y_t \qquad \forall t \in A \qquad (5)$$

Constraint (5) ensures that AMLs can perform their tours only if they are assigned.

$$\sum_{j \in N / N_C} u_{ji} - \sum_{j \in N / N_C} u_{ij} = \sum_{j \in N_C} u_{ij} \qquad \forall i \in N_V \qquad (6)$$

$$\sum_{j \in N / N_C} v_{ij} - \sum_{j \in N / N_C} v_{ji} = \sum_{j \in N_C} v_{ji} \qquad \forall i \in N_V \qquad (7)$$

Constraints (6) and (7) calculate the pick-up and delivery loads at the stops. Constraints (6) and (7) act as sub-tour eliminations equations as well.

$$u_{ij} + v_{ij} \leq CV \sum_{t \in A} x_{ijt} \qquad \forall i, j \in N: i, j \notin N_C, (i,j) \notin N_0, i \neq j \qquad (8)$$

Constraints (8) that the total load carried by AMLs do not exceed their respective capacities.

$$u_{ij} \leq CV - \sum_{k \in N_C} u_{ki} + M\left(1 - \sum_{t \in A} x_{ijt}\right) \qquad \forall i \in N_V, j \in N: j \notin N_C, i \neq j \qquad (9)$$

$$v_{ij} \le CV - \sum_{k \in N_C} v_{ki} + M\left(1 - \sum_{t \in A} x_{ijt}\right) \qquad \forall i \in N_V, j \in N: j \notin N_C, i \neq j \tag{10}$$

Constraints (9) and (10) calculate the remaining capacity of AMLs when carrying delivery and pickup loads, respectively.

$$u_{ij} \ge \sum_{k \in N_C} u_{jk} - M\left(1 - \sum_{t \in A} x_{ijt}\right) \qquad \forall i \in N_V, j \in N/N_C, i \neq j \tag{11}$$

$$v_{ij} \ge \sum_{k \in N_C} v_{ki} - M\left(1 - \sum_{t \in A} x_{ijt}\right) \qquad \forall i \in N_V, j \in N/N_C, i \neq j \tag{12}$$

Constraints (11) and (12) calculate the total load of deliveries and pickups carried by AMLs.

$$\sum_{i \in N_V} u_{ki} \le CD_k \qquad \forall k \in N_0 \tag{13}$$

$$\sum_{i \in N_V} v_{ik} \le CD_k \qquad \forall k \in N_0 \tag{14}$$

Constraints (13) and (14) put a limitation on the depot capacity, ensuring that the total pickup and delivery demands assigned to the depot are within this limit.

$$\sum_{k \in N_0} x_{ikt} \le q_{it} \qquad \forall i \in N_V, t \in A \tag{15}$$

$$\sum_{k \in N_0} x_{kit} \le q_{it} \qquad \forall i \in N_V, t \in A \tag{16}$$

Constraints (15) guarantee that AMLs cannot visit any stop unless they have been assigned to a depot. Constraints (16) guarantee that each AML can only end a tour to the depot from which it has been assigned.

$$\sum_{i,j \in N, i,j \notin N_C, i \neq j, (i.j) \notin N_0} dist_{ij} x_{ijt} \le rv \qquad \forall t \in A \tag{17}$$

Constraint (17) ensures that the length of tours carried out by each AML does not exceed a predetermined range. Note that the allowed range associated with AMLs is based on their battery capacity.

$$h_{jt} - 1 \geq h_{it} - |N_V|\big(1 - x_{ijt}\big) \qquad \forall i \in N_V, j \in N_V, t \in A, (i,j,t) \in \mathcal{L} \tag{18}$$

$$h_{jt} \geq 1 - |N_V|\big(1 - x_{ijt}\big) \qquad \forall i \in N_0, j \in N_V, t \in A, (i,j,t) \in \mathcal{L} \tag{19}$$

Constraints (18) and (19) by labelling each node on the AML routes function as subtour elimination constraints.

The second echelon constraints:

$$\sum_{j \in N, t \in C, (i,j,t) \in \mathcal{F}} x_{jit} = 1 \qquad \forall i \in N_C \tag{20}$$

Constraint (20) ensures each customer node is visited by a courier once and only once.

$$\sum_{i \in N_V, j \in N/N_0, (i,j,t) \in \mathcal{F}} x_{ijt} \leq |N_V| y_t \qquad \forall t \in C \tag{21}$$

$$\sum_{j \in N/N_0, i \neq j} x_{jit} \leq \sum_{j \in N/N_0, i \neq j} x_{ijt} \qquad \forall t \in C, i \in N_C \tag{22}$$

Constraint (21) ensures that couriers can start their mission once they are employed. Constraint (22) forces the couriers to leave the customer locations after performing their missions.

$$\sum_{j \in N/N_0} u_{ji} - \sum_{j \in N/N_0} u_{ij} = d_i \qquad \forall i \in N_C \tag{23}$$

$$\sum_{j \in N/N_0} v_{ij} - \sum_{j \in N/N_0} v_{ji} = p_i \qquad \forall i \in N_C \tag{24}$$

Constraints (23) and (24) guarantee that all pickup and delivery demands are met.

$$u_{ij} + v_{ij} \leq CC \sum_{t \in C} x_{ijt} \qquad \forall i, j \in N_C : i \neq j \tag{25}$$

Constraint (25) ensures that the total load carried by couriers do not exceed their respective capacities.

$$u_{ij} \leq CC \sum_{t \in C} x_{ijt} \qquad \forall i \in N_V, j \in N_C \tag{26}$$

Constraint (26) limits the load of pickups and deliveries to the courier's capacity.

$$v_{ij} \leq CC \sum_{t \in C} x_{ijt} \qquad \forall i \in N_C, j \in N_V \tag{27}$$

$$\sum_{j \in N_C} u_{ij} = \sum_{j \in N_C} z_{ji} d_j \qquad \forall i \in N_V \tag{28}$$

$$\sum_{j \in N_C} v_{ji} = \sum_{j \in N_C} f_{ji} p_j \qquad \forall i \in N_V \tag{29}$$

Constraint (27) limits the load of pickups and deliveries to the courier's capacity. Note that couriers carry only pickups when arriving at stops having made all their deliveries. Similarly, couriers carry only deliveries when leaving a stop after being supplied by an AML. Constraints (28) and (29) ensure that the total pickup and delivery loads carried by couriers must be equal to customer demands assigned to their corresponding stops.

$$u_{ij} \leq (CC - d_i) \sum_{t \in C} x_{ijt} \qquad \forall i \in N_C, j \in N: j \notin N_0, i \neq j \tag{30}$$

$$v_{ij} \leq (CC - p_i) \sum_{t \in C} x_{ijt} \qquad \forall i \in N_C, j \in N: j \notin N_0, i \neq j \tag{31}$$

Constraints (30) and (31) ensure that delivery and pickup operations can be fulfilled by the couriers provided that sufficient capacity is available.

$$u_{ij} \geq d_j \sum_{t \in C} x_{ijt} \qquad \forall i \in N_C, j \in N_C: j \notin N_0, i \neq j \tag{32}$$

$$v_{ij} \geq p_i \sum_{t \in C} x_{ijc} \qquad \forall i \in N_C, j \in N_C: j \notin N_0, i \neq j \tag{33}$$

Constraints (32) and (33) ensure that all delivery and pickup demands are met.

$$\sum_{j \in N_V} z_{ij} = 1 \qquad \forall i \in N_C \tag{34}$$

$$\sum_{j \in N_V} f_{ij} = 1 \qquad \forall i \in N_C \tag{35}$$

Constraints (34) and (35) ensure that each customer is assigned to a unique stop.

$$\sum_{i,j \in N, i,j \notin N_0, i \neq j, (i.j) \notin N_V} dist_{ij} x_{ijt} \leq rc \qquad \forall t \in C \tag{36}$$

Constraint (36) ensures that the length of tours carried out by courier does not exceed a predetermined range.

$$\sum_{j \in N_C} x_{ijt} \leq \sum_{j \in N_V, (i,j,t) \in \mathcal{F}} x_{jit} + |N_V|(s_{it} + b_{it}) \qquad \forall i \in N_V, t \in C \tag{37}$$

$$\sum_{j \in N_C} x_{jit} \leq \sum_{j \in N_V, (i,j,t) \in \mathcal{F}} x_{ijt} + |N_V|(e_{it} + b_{it}) \qquad \forall i \in N_V, t \in C \tag{38}$$

$$\sum_{j \in N_V, (i,j,t) \in \mathcal{F}} x_{ijt} = \sum_{j \in N_V, (i,j,t) \in \mathcal{F}} x_{jit} \qquad \forall i \in N_V, t \in C \tag{39}$$

Constraint (37) states that a courier can leave a stop to make a tour when a courier has either reached that stop from another stop or that stop is the starting point. Constraint (38) guarantees that a courier will return to a stop to make another tour or to end its mission. Constraint (39) ensures courier conservation at the stops in the second echelon.

$$|N_V|(2 - s_{it} - e_{it}) + |N_V|b_{it} + |N_V| \sum_{j \in N_V, (i,j,t) \in \mathcal{F}} x_{ijt} \geq \sum_{j \in N_V, k \in N_C, (j,k,t) \in \mathcal{F}, i \neq j} x_{jkt} \qquad \forall i \in N_V, t \in C \tag{40}$$

Constraint (40) states that couriers are allowed to visit several stops across their journey only in the following cases:

the start and end stops are not the same,

an open mission is performed, and

the next stop is another stop.

$$\sum_{i \in N_V} s_{it} = y_t \qquad \forall t \in C \tag{41}$$

$$\sum_{i \in N_V} e_{it} = y_t \qquad \forall t \in C \tag{42}$$

Constraints (41) and (42) ensure that if a courier is employed, he is assigned to a particular start and end stop.

$$w_{ji}^1 - w_{ji}^2 = f_{ji} - z_{ji} \qquad \forall i \in N_V, j \in N_C \tag{43}$$

$$w_{ji}^1 + w_{ji}^2 \leq 2 - f_{ji} - z_{ji} \qquad \forall i \in N_V, j \in N_C \tag{44}$$

$$w_{ji}^1 + w_{ji}^2 \leq f_{ji} + z_{ji} \qquad \forall i \in N_V, j \in N_C \qquad (45)$$

Constraints (43)-(45) force w_{ji}^1 and w_{ji}^2 to be zero when both the delivery and pickup tasks of costumer $j \in N_C$ is assigned to stop $i \in N_V$. Note that f_{ji} and z_{ji} can be unity at the same time, meaning that both the delivery and pickup demands for customer $j \in N_C$ can be assigned to stop $i \in N_V$.

$$b_{it} \leq \sum_{j \in N_C} w_{ji}^1 + \sum_{j \in N_C} w_{ji}^2 \qquad \forall i \in N_V, t \in C \qquad (46)$$

Constraint (46) inhibits a courier from performing an open mission from a stop to which customers with both pickup and delivery demands are assigned as the courier must return to that stop in order to supply the AML with the pickups.

2.5 AN ADAPTIVE BACKTRACKING-SA MATHEURISTIC

Two-echelon VRPs are regarded as NP-hard owing to their nature. This issue becomes even more complex when operational assumptions are added to the original problem. The search space of the problem is all combinations of courier and AML routes, restricted by various constraints, including the maximum load capacity, maximum travel distance, and the conservation of couriers and AMLs. All these constraints lead to the NP-hard nature of the 2-EVRP.

The selected method for addressing the 2E-MOVRP problem follows a two-stage approach, incorporating the Backtracking Search Algorithm (BTA) and Simulated Annealing (SA) algorithms. This choice is grounded in several key considerations:

- The use of both BTA and SA for the courier routes serves to strike a balance between generating efficient initial routes and further refining them to achieve better solutions. BTA rapidly provides feasible initial solutions, while SA fine-tunes these solutions to optimize the courier routes.
- Employing BTA for the AML routes is a well-suited choice, given the relatively small size of the first echelon network, which primarily consists of connections between depots and stops. BTA can efficiently and swiftly generate all required routes within this network, encompassing both open and closed routes. This eliminates the need for additional time-consuming optimization through SA.

- The proposed approach closely mirrors practical logistics scenarios. In real-world logistics operations, the routing of couriers and AMLs often occurs in stages. This approach aligns with this reality, where courier routes are initially planned and subsequently used as a basis for finalizing AML routes, taking into account the identified stops and courier routes.

This section presents an effective method to overcome the problem's computational complexity, exploring the feasible region by splitting the problem into subproblems.

2.5.1 General structure

The algorithm presented in this study solves the 2E-MOVRP in two stages. In the first stage, the courier routes are initially determined and in the next stage, according to the routes generated in the first stage, the AML routes are constructed. Both of these stages include several steps. Figure 2.3 provides a schematic view of the overall algorithm.

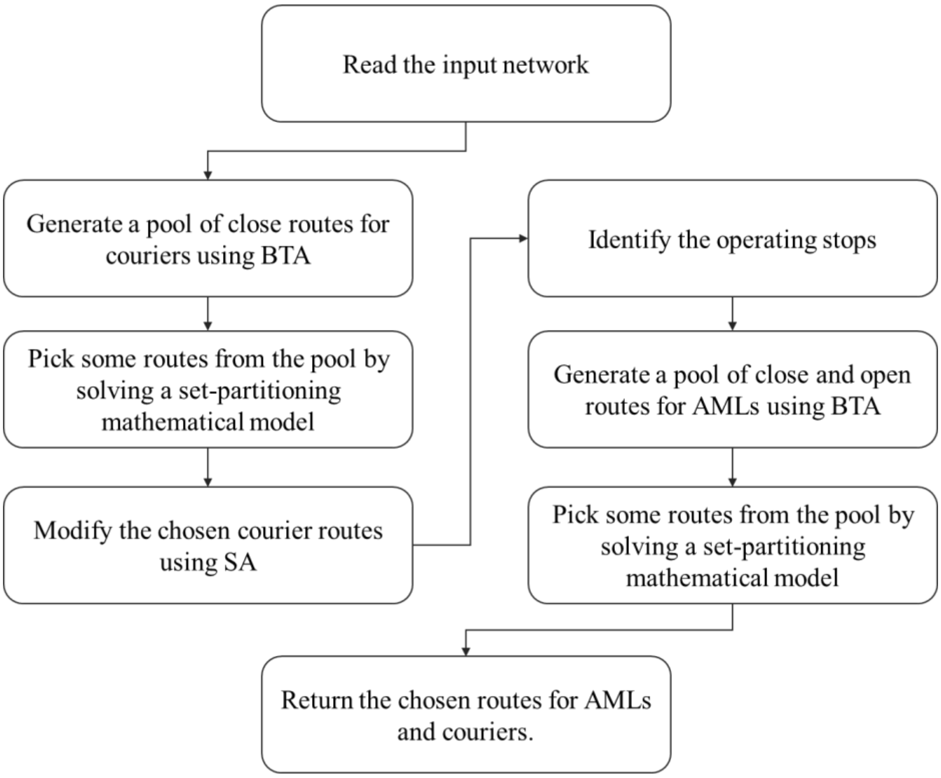


Figure 2.3. The overall schematic of the proposed solution approach.

To generate the courier routes three consecutive steps are carried out. In the first step, we use the Backtracking Search Algorithm (BTA) to generate closed routes, stored in a pool of routes. In the subsequent step, a set-partitioning problem selects some of these routes with the aim of minimizing the operating cost of the couriers. The result will be a feasible initial solution with closed routes. In the third step, the initial solution is improved by applying the Simulated Annealing (SA) algorithm to construct the final courier routes.

To generate AML routes, the stops included in the courier routes are first identified. Then, through a process similar to that used to generate courier routes, a pool of routes is initially generated by the BTA. Afterwards, the set-partitioning problem is solved to search for the best combination of the routes with the objective of minimizing the total cost. Due to the small size of the first echelon network (which includes only links between depots and stops), the BTA, can generate all open and closed routes in a short time, whose origins and destinations are depots. Therefore, the solution obtained from the mathematical model is not required to be improved further and can be considered as the final AML routes.

2.5.2 Backtracking search algorithm to generate the initial solution

The search space for 2E-MOVRP is the set of all routes that includes all links between customers and stops in the second echelon and between depots and stops in the first echelon. The search space is restricted by various constraints including the maximum payload capacity and the maximum length of travel for both couriers and AMLs as well as the characteristics of the pickup and delivery demands.

The ABSM algorithm first uses the BTA to generate the initial solution. The BTA is extensively used in different studies to solve combinatorial optimization problems (Civicioglu, 2013; Tarantilis et al., 2002; Tarantilis et al., 2004; Akhtar et al., 2017). The advantage of the backtracking method is that some states can be excluded without having to examine all the search space. This algorithm is usually implemented in the form of recursive functions.

The BTA used in this study builds routes using an adjacency matrix, augmented by several innovative processes developed in this chapter to reduce the search space. The adjacency matrix of an undirected network graph (containing no loops) is a symmetric matrix whose rows and columns are indexed by the node numbers. In this matrix, the *ij*-entry takes 1 if there is a direct connection from node *i* to node *j* and 0 otherwise. According to this matrix, different paths can be obtained that connect each pair of nodes existing in the network. However, since there is no physical link between the nodes, all nodes can be perceived as adjacent to each other. As a result, a large number of possible paths can be generated between two nodes, even for a network with a relatively small size. Therefore, in terms of computational effort and the number of calculations, it is not possible to examine all combinations for large networks. As a result, in this paper, for a particular node, k-nearby nodes are considered as adjacent nodes. However, considering the same value of k for all networks of large sizes still takes considerable computational effort since a considerable number of routes are generated. Note that a large value of k for the nodes that stand a far distance from other points only leads to the production of long paths that are unlikely to be present in the optimal solution. Therefore, in this paper, a novel adaptive procedure for calculating k per node is suggested and elaborated below.

2.5.3 Reducing the size of search space

In the proposed solution method, we consider two techniques to reduce the search space. First, *K-nearest neighbourhood search* method helps the algorithm to avoid the extensive search of all the neighbourhoods of a node, which in return shrinks the search space and reduces the solution time. Second, *load to length ratio (LLR)* policy reduces the search space further by excluding routes with extensive length and inefficient use of couriers' capacity.

Adaptive K-nearest neighbourhood search

To build the adjacency matrix described above, first, the average distances for all nodes of the network are calculated using the following formula:

$$AvgDist = \frac{\sum_{i \in N} \sum_{j \in N} dist_{ij}}{|N| \times (|N| - 1)}$$

We introduce δ as the adjustment coefficient. By considering the initial value of 1 for δ, the number of neighbours located within the $\frac{AvgDist}{\delta}$ range is determined for each node of the network. If the average number of these neighbours is greater than a predetermined value (we call it *k*), we increase the value of δ (in this study the value of 0.01 is considered). This continues until the number of adjacent nodes for all network nodes is less than *k*. Then, for nodes with more than k adjacent nodes, the number of adjacent nodes is adjusted to k. Also, for nodes whose number of adjacent nodes is less than 2, the value of k is considered equal to 2.

Load to length ratio

Theoretically, all possible paths between each OD pair have the potential to be generated. However, it does not make sense to choose some routes due to their excessive length with inefficient use of couriers' capacity. Therefore, it is unlikely that such routes will appear in any optimal or near-optimal solution. In this paper, the concept of *Load to Length Ratio* (LLR) is introduced to reduce the size of the set of routes. The higher the value of this indicator, the shorter the paths produced by the more efficient use of the couriers' capacity. The following statement shows how to calculate the LLR of each path *p*:

$$LLR = \frac{\left(\frac{deliveryload + pickupload}{2 \times cc}\right)}{\left(\frac{length}{rc}\right)}$$

where rc and cc represent the maximum length of a courier path and the maximum load capacity of a courier, respectively.

During the path generation process, the routes are evaluated by the *LLR* measure and the path that fails to meet this criterion is truncated at this stage. Each route contains a sequence of nodes visited along the path. Hence, each route is unique, even if there is more than one route between a given OD pair. In Appendix 2.A, Algorithm 2.1 summarizes the BTA used to generate the pool of routes for the second echelon. The BTA for the first echelon is similar.

One of the parameters used in the BTA is the $CloseNeighbors$ matrix, which is set according to the network topology. The $CloseNeighbors$ matrix is a matrix with the dimension of $|N| \times |N-1|$ where the columns of row *i* represent the nodes sorted in order of the nearest to the farthest distance from the other nodes. (e.g., $Closeneighbour_i(1)$) is the closest node to node *i*). Then, according to the *adaptive-k-nearest-neighbour* method, the number of neighbours ($MaxNeighbor$) to be searched in the BTA for each point is determined. Since some points may be farther away than other nodes of the network, they may not be visited by routes inclusive of their immediate neighbours due to the limited carrying capacity of couriers. Therefore, it is necessary to consider a wider neighbourhood of origin for the production stage of routes when the BTA starts the truncating process. Therefore, for the origins only (which are AML stops (N_V)), the $MaxNeighbor$ value is considered to be a number greater than the value obtained from the *adaptive-k-nearest-neighbour ($StopK$)* method.

The BTA uses a list of visited nodes (*Vs*) that takes *True* for nodes included in a route generated and *False* for nodes that have not yet been visited. Depending on the echelon whose pool of routes is supposed to be generated by the BTA, the list of nodes visited for another echelon will be $True$. This strategy prevents cycling back to the nodes which are already visited. In the next step, the destination node and the origin node are specified. At the beginning, the current node and the previous node are set to the origin node, and then, the path generation process initiates.

The $GeneratePath$ function is a recursive function that generates routes by forming a search tree and branching in the neighbourhood of each node. Since all problem constraints are checked during the route production process, the routes produced are all feasible. Finally, the routes generated are stored in the $AllPath$ set for a specific OD pair.

The BTA creates all closed and open routes between depots and stops to create a set of first-echelon routes and stores them in the corresponding pool. However, for the second echelon, which is larger and denser than the first echelon, the BTA creates only a set of closed routes, starting from a stop and ending at the same stop. Section 5.5 explains how open routes are built in the second echelon.

2.5.4 Set partitioning formulation to select the best initial routes

After generating a pool of routes, using an integer programming (IP) model, the routes with the lowest cost are selected. For each echelon, the corresponding MIP is applied. To generate the initial routes of the courier, the IP model seeks to minimize the operating cost of the couriers considering the AML's capacity. Constraints (60) to (64) represent the IP formulated for the second echelon.

Sets

$\wp$: The set of courier paths generated by Backtracking algorithm

Parameters

a_{ip}: A binary parameter which equals to 1 if path $p \in \wp$ visits costumer $i \in N_C$.

c^c: Courier operation cost

$Ds(p)$: The destination of path $p \in \wp$

$O(p)$: The origin of path $p \in \wp$

Dl_p: The total deliver load of path $p \in \wp$

Pk_p: The total pickup load of path $p \in \wp$

l_p: The length of path $p \in \wp$

cv: The maximum capacity of AMLs

Variables

R_p: A binary variable which takes 1 when path $p \in \wp$ is chosen.

$$Min \sum_{p \in \wp} l_p c^c R_p \tag{47}$$

$$\sum_{p \in \wp} a_{ip} R_p = 1 \qquad \forall i \in N_C \tag{48}$$

$$\sum_{p \in \wp : O(p)=i} Dl_p R_p \leq cv \qquad \forall i \in N_V \tag{49}$$

$$\sum_{p \in \wp : D(p)=i} Pk_p R_p \leq cv \qquad \forall i \in N_V \tag{50}$$

$$R_p \in \{0,1\} \qquad p \in \wp \tag{51}$$

Objective (47) seeks to minimize the operating costs associated with courier routes. Constraint (48) together with (47) selects the path with the lowest cost. Constraints (49) and (50) guarantee that the delivery and pickup loads do not exceed AML capacity. Constraint (51) specifies the domain of the binary variable.

Objective (52) with constraints (53) to (55) represent the model for the selection of routes of the AMLs, which yield the minimum overall cost. Definitions for the Sets, Parameters and Variables used are given below.

Sets

$\wp'$: The set of AML paths generated by the BTA.

N'_V: The set of stops that are operated due to the courier routes.

Parameters

a'_{ip}: A binary parameter which equals to 1 if path $p' \in \wp'$ visits stop $i \in N_V$.

l'_p: the length of path $p' \in \wp'$

c^a: AML operation cost.

Variables

R'_p:	A binary variable which takes 1 when path $p \in \wp$ is chosen.

$$Min \sum_{p\in\wp} \left(l'_p c^a + f^V\right) R'_p \tag{52}$$

$$\sum_{p\in\wp} a'_{ip} R'_p = 1 \qquad \forall i \in N'_V \tag{53}$$

$$\sum_{p\in\wp:O(p)=i} R'_p - \sum_{p\in\wp:D(p)=i} R'_p = 0 \qquad \forall i \in N_D \tag{54}$$

$$R'_p \in \{0,1\} \qquad p \in \wp' \tag{55}$$

2.5.5 Simulated Annealing (SA) metaheuristic for constructing courier routes

The Simulated Annealing algorithm (SA) was independently introduced by Kirkpatrick et al. (1987) and Černý (1985). The study conducted by Metropolis et al. (1953) in the field of Statistical Mechanics laid the theoretical foundation for the SA heuristic. Since then, SA has been extensively applied to solve various discrete combinatorial optimization problems (Van Breedam, 1995);(Jayaraman & Ross, 2003)03;(Wei et al., 2018); (S.-W. Lin et al., 2009).

Since the initial solution for courier routes in the second echelon obtained by the BTA do not include open routes, another process is required to investigate the search space more broadly while still covering the problem assumptions. In the proposed solution method, the SA algorithm is used to modify the initial solution and construct open routes. Note that the SA algorithm presented in this study adjusts the parameters in order to survey the search space more efficiently. Algorithm 2.2, in the Appendix A, details the steps of the proposed SA algorithm. The parameters used in the proposed algorithm and their description are as follow:

Parameter	Description
$nMove$	Number of moves made for swapping the costumers
$Temp$	The temperature used in SA
$MinTemp$	The minimum temperature in SA
$EquibliriumTenure(ET)$	The number of no-improvement iterations which establishes equilibrium condition

$DistRate(DR)$	The ratio which determines the maximum distance between two swapped costumers
$LoadToLenghtRatio$	The load –to-length ratio
$Max_{NoImprove}$	The maximum number of iterations with no improvement
k	The depth of BT search tree
$StopK(SK)$	The number of adjacent nodes for stops.

First, the initial values of the algorithm parameters are set. The conditions for termination of the algorithm are that $Temp$ reaches a value less than $MinTemp$ or the algorithm does not improve the current solution as much as $MaxNoImprove$ iterations. Until one of the termination conditions is met, the algorithm generates a new solution at each iteration with respect to its current solution in the search space by calling the $MoveToNeighbor$ function. This function is described in detail in Section 5.5.1. If the best solution so far is found, it will be updated to the new answer, otherwise, it will be checked whether this solution leads to an improvement in the current solution or not. If improved, the current solution will be updated to the new one; otherwise, the current solution, which is a worse solution, will potentially be accepted and updated with probability of $exp\left(\frac{-Obj_{NewSol}-Obj_{CurrentSol}}{Temp}\right)$. This process of accepting a worse result during the process is called the Metropolis criteria, which helps to prevent early stagnation (Metropolis et al., 1953).

If the algorithm cannot improve the best solution found so far in ET consecutive iterations, the equilibrium condition is established. If the equilibrium condition is met, the parameters of the $MoveToNeighbor$ function are adopted for a more extensive search. Whereas, if the equilibrium condition is not met, the parameters of the $MoveToNeighbor$ function are adjusted for a more restricted search. This mechanism helps the algorithm accelerate the search process. With these updates, the algorithm sets a balance between execution time and solution quality.

Neighbourhood moves

Each solution generated by the SA algorithm consists of three types of routes; courier routes, AML routes, and merged courier routes (multiple routes of courier). To move from the current solution to a neighbouring answer, a novel algorithm called $MoveToNeighbor$ has been developed. In this process, first, all possible moves are specified. Then, the one leading to the lowest cost is selected as the neighbouring solution. The routes in the new solution are then examined to create open and merged routes. Figure 2.4 shows the flowchart of the $MoveToNeighbor$ function.

Possible neighbourhood moves

In order to determine the feasible moves, in the first step, the distances between each pair of customers in separate routes are calculated. These distances are then normalized by dividing by the maximum distance (shown by $dist_{ij}^N$). Afterwards, the possible reassignment of two customer nodes in two separate routes are assessed. To do so, we introduce the $DistRate$ indicator whereby the possible reassignments are identified. The logic behind this innovation is that reassigning two customer nodes to separate routes that are far away from each other is unlikely to lead to an improvement in the current solution. Therefore, in the second step, the reassignment of nodes with normalized distances greater than $DistRate$ are ignored. These two steps are shown as follows, where R represents the set of all courier routes in the current solution:

Step 1: $$dist_{ij}^N = \frac{dist_{ij}}{\max\limits_{\forall k \in R^1, \forall m \in R^2} \{dist_{km}\}} \quad \forall R^1, R^2 \in R: R^1 \neq R^2, \forall i \in R^1, \forall j \in R^2$$

Step 2: $$Moves = \left\{(i,j) \middle| dist_{ij}^N \leq DistRate \forall R^1, R^2 \in R: R^1 \neq R^2, \forall i \in R^1, \forall j \in R^2\right\}$$

Move selection mechanism

After determining the possible moves to neighbouring solutions, a number of these moves are then selected according to a roulette wheel mechanism. Two commonly procedures distinguished in the literature are applied in this study:

- First, removing a customer node from a route and adding it to another route.
- Second, exchanging two customers nodes between routes.

Figure 2.5 shows the strategies applied to generate a neighbouring solution.

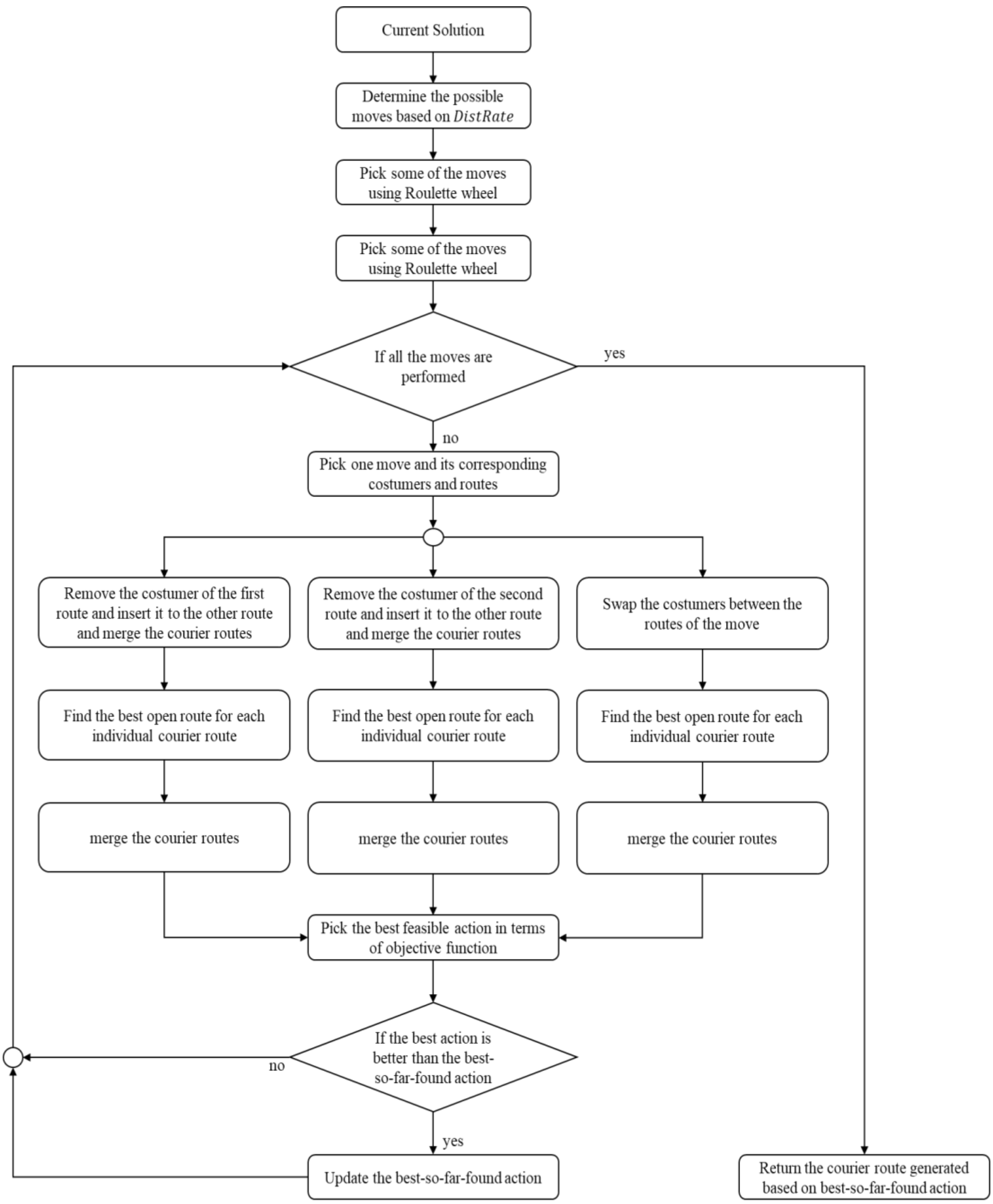
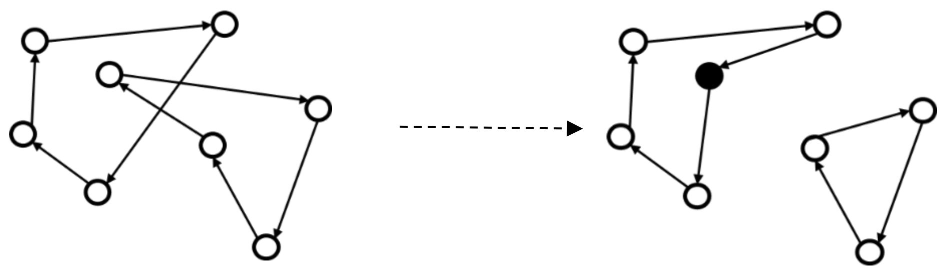


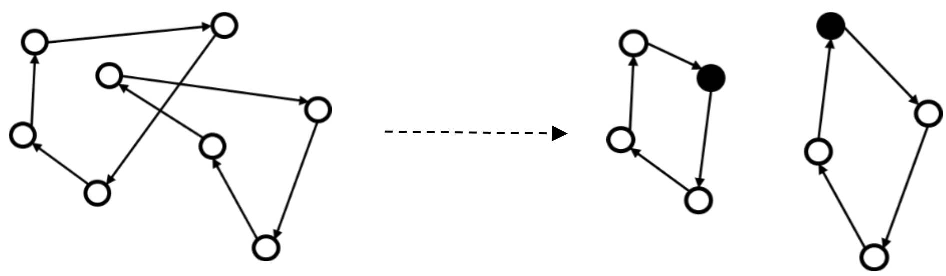
Figure 2.4. The flowchart of the $MoveToNeighbor$ function

As each node is added to another route, the new route is modified using the BTA. The adjacency matrix in this step is then quantified with 1 for all nodes in the route and with 0 for other nodes. Also, instead of the LLR ratio, the shortest route so far achieved is used to prevent extra branching. If the route improves the shortest path found so far, the values for the shortest route are updated accordingly. The output of this step will be the shortest route that is considered as an improved route. This algorithm also checks the feasibility when generating an improved route.

As a result, we are finally ensured that the improved route is feasible for the given network. If the move is not feasible, the move will be considered null-and-void. After making open and merged courier routes, the operating cost and fixed cost of the courier routes are calculated. From all the moves made, each one that leads to the lowest cost is selected as the neighbour solution. Figure 2.5 shows two strategies of moving to a neighbouring solution schematically.



Removing a customer node from a route and adding it to another route



Exchanging two customers nodes between routes

Figure 2.5. Two strategies of moving to a neighbouring solution

Constructing open routes

Since it is assumed that the courier can return to a stop other than its origin stop, in this step we assign the origin and destination of each route $r \in R$ to its nearest stop. This replaces the routes where the courier travels a long distance to make closed routes with a shorter open route. One of the unfortunate consequences of this move is that the courier conservation constraint may be violated. Investigating this issue and resolving it will be reviewed when merging routes.

Merging courier routes to make multi-mission paths

As the last step in the production of courier routes, we merge the previously generated routes to implement the multi-mission assumption for courier routes. In this study, a novel process is designed that can efficiently merge routes and provide an acceptable solution. When merging the routes, the courier conservation constraint must also hold, and if not, the routes are modified accordingly. Algorithm 2.3 (see Appendix 2.A) describes in detail how routes are merged.

The merging process will be applied on the courier routes corresponding to a given solution. Let $\mathfrak{R}^{new}$ denote the new set of courier routes. The new set is empty in the first place and meant to encompass merged routes. At the beginning, one of the routes is selected as the reference route. Then, the route whose origin or destination is closest to the destination of the reference route is selected. If the origin is closer, we append the route to the end of the reference route without making any changes. Otherwise, if the destination is closer, we reverse the route and check its feasibility. If feasible, we append the route to the reference route. After merging, we check whether the destination of the reference route or the origin of the second route is better suited as the connection point for these two routes. The connection point leading to the shorter distance is selected, and the reference route is updated accordingly. If the reference route becomes too long for the courier, the second route will not be appended to the reference route. In this case, the reference route is added to $\mathfrak{R}^{new}$. The new route is selected as the reference route, and all the described processes will continue until no further merging operation is feasible. Each merged route represents the total missions a courier makes.

Each of the routes in $\mathfrak{R}$ is set once as the reference route and the whole process described above is repeated. In order to be able to effectively examine the different practices of merging routes, we once again consider the reverse of each of the routes in $\mathfrak{R}$ as the reference route and repeat the process of merging the routes. Finally, for each of the merged routes, we check the conservation constraint of couriers at the stops. If the last stop of each route $r \in \mathfrak{R}^{new}$ is not the origin stop, we change the last stop to the origin stop. During this process, the $\mathfrak{R}^{new}$constructed by the different techniques are evaluated and the one which has the lowest operating and fixed cost is identified as the final courier route ($\mathfrak{R}^{best}$).

Adaptive search method

The $DistRate$ and $nMoves$ parameters have a significant impact on the searching mechanism of the algorithm. Setting these parameters to a certain value makes the algorithm not be able to effectively search the solution space and produce quality solutions. Moreover, it causes the algorithm to make unnecessary iterations, which in turn increases the solution time dramatically.

In this study, an equilibrium condition is considered that constantly monitors the process of improving the algorithm and adjust the $DistRate$ and $nMoves$ values. If the algorithm fails to improve the best solution found for a number of predetermined iterations, the equilibrium condition is established. As a result, the number of $nMoves$ and the value of $DistRate$ need to be increased in order to search the solution space more extensively. On the other hand, when there is no equilibrium condition, the algorithm reduces the number of $nMoves$ and $DistRate$ to accelerate the solving process. To adjust the values of $DistRate$ and $nMoves$ coefficient $0 \leq \alpha \leq 1$ for $nMoves$ and $0 \leq \beta \leq 1$ for $DistRate$ are considered. If the equilibrium condition is met for an extensive search, the values of $nMoves$ and $DistRate$ are each multiplied by $(1 + \alpha)$ and $(1 + \beta)$, respectively. On the other hand, in the case of non-equilibrium conditions to shrink the search space, each $nMoves$ and $DistRate$ are multiplied by $(1 - \alpha\ )$ and $(1 - \beta)$, respectively. It is to be noted that $nMoves$ must be greater than ten and that $DistRate$ must be less than 1.

The computational experiments on the algorithm's parameters are provided in Appendix B.

2.6 RESULTS

In this section, the MILP model provided in Section 2.4 and the heuristic developed to solve the problem are examined and analysed via several numerical experiments. Since no existing testbed is available in the literature addressing the 2E-MOVRP, we first elaborate how to generate the instances in Section 2.5.1. Then, in Section 2.5.2, the performance of the algorithm is examined by comparing the quality of solutions and runtime with the solutions rendered by a standard MILP solver.

All our procedures were implemented in Java (version SE 11) and executed on an x64-PC with an Intel Core i7-7600 1.3 GHz CPU and 16.0 GB of RAM. We used Gurobi version 9.0.2 as the MILP solver.

2.6.1 Instance generation

Since the AML-courier delivery system is not yet widely applied in practice and is in the early stages of operation, accurate extraction of the required parameters may not be possible. Therefore, all parameters have either been extracted from the relevant literature, estimates made by the authors and/or subject to sensitivity analysis.

As a logical principle when designing a parcel distribution network, facilities should be scattered throughout the network, and each covers a certain area. Therefore, in the phase of network generation, attention has been paid to the fact that stops and depots should be located at a suitable distance from customers and cover the entire study area. Geographic information of the customer locations and depots are randomly generated. The Manhattan distance metric is used to calculate the distances in acknowledgement of the existence of a network of roads, cycle lanes, pedestrian precincts, and walkways.

To generate the pickup and delivery demands, a uniform distribution function is used in a range of 1 kg to 50 kg to ensure all parcels can be carried by couriers. It is assumed that the couriers can travel a total distance of 30 km per day and carry packages with a maximum weight of 50 kg. Each AML has a maximum payload capacity of 500 kg, which can travel a maximum of 10 km per day after their batteries are fully charged at depots. We consider a fixed cost of \$50 for using AMLs, based on the daily rental cost. The operating cost of AMLs is calculated according to the electricity consumed per kilometre, which is \$1/km. Fixed courier costs can include set fees for long term service contracts such as payroll taxes and a base salary paid by a company, which is considered to be \$40 per day. In addition to the base salary, couriers get paid based on the kilometrage, at the rate of \$4 per kilometre. The aforementioned cost parameters have been obtained through interviews with subject experts and reports. However, it should be noted that these parameters may vary depending on the specific location where the operations are conducted.

In order to avoid biased data, different network configurations are randomly generated, each of which differs in terms of the number of depots, stops, customers, and grid dimension. The grid area (a square) varies from 1 km^2 to 36 km^2, with side length between 1km and 6 km.

2.6.2 Evaluation of algorithm performance

In this section, the performance of the proposed algorithm in compared with the MILP solver on small networks. Each grid is run 10 times by the algorithm and the results consisting of the best solution (Best) and the averaged CPU time (sec) are shown in Table 2.7 (Appendix 2.C). The performance of the algorithm is compared with the Gurobi solver obtained by solving the MILP model according to the two measures of execution time (sec) and optimality gap (%dev). The geographical information of small instances is randomly generated with 10, 15 and 20 customers in a grid with area ($|\omega|$) 1 km^2. Since solving the MILP model for some instances is very costly in terms of computational time, a maximum of 1800 seconds is set as the maximum CPU time for Gurobi. The best solution obtained for the MILP solver after 1800 s appears in the 'Best' column of Table 2.7. Also, as optimality has not been achieved within 1800 seconds for some network configurations, the objective is likely to yield a solution with an optimality gap, for which the lower bounds (LB) are also reported in Table 2.7 (see Appendix C).

Table 2.7 shows that the proposed heuristic is capable of converging to the optimal solution in most instances. In the networks with 10 customers (A1-A20), the heuristic could converge to the optimal solution. The average optimality gap for instances with 15 customers (B1-B20) is 0.44% with an average of 1 second CPU time. Since the Gurobi solver is not able to converge to the optimal solution within 1800 (s) for most networks with 20 customers (C1-C20), Best, representing the upper bound, and LB, representing the lower bond, are reported. The optimality gap between the algorithm's objective and the upper bound (Best) is shown with dev (%). As seen in Table 2.7, in some cases, the algorithm results in a better solution than the upper bound obtained, for which the optimality gap is reported with a negative value.

From the results for the network instances, a sharp increase in the execution time of the MIP model is perceived for a slight increase in the number of nodes. The average required CPU time for Gurobi to solve networks A1-A20 and B1-B20 are 109.163 and 724.4715 seconds, respectively. However, the heuristic method is much faster for increasing problem size, which for none of the cases presented in Table 2.7 exceeds 1 second.

2.6.3 Efficiency test for the medium size instances

In order to evaluate the performance of the proposed heuristics further, the MILP model is fed the heuristic solution as its starting point to investigate whether the MILP can improve the initial solution within a maximum allowed run-time of 1800 seconds. In doing so, 10 random network instances with 2 depots, 3 stops and 50 customers are constructed randomly. As reported in Table 2.8 (Appendix C), after 1800 seconds, only in one case could the Gurobi solver improve the heuristic solution. Note that 1800 seconds of run-time is not the time it takes for Gurobi to solve the problem, rather this is the amount of time Gurobi spends trying to improve the heuristic solution. When the runtime reaches its limit, some solutions have not converged. UB is the best solution that Gurobi reaches within 1800 (s).

2.6.4 Comparing 2E-MOVRP with 2E-VRP, 2E-MVRP and 'courier-only' for large networks

In this section, we examine the cost savings obtained by considering open routes and multi-trip operations. As previously mentioned, this study is an extension of Jun Li et al. (2021), in which a two-echelon pickup and delivery routing problem is addressed. In the earlier work it was assumed that couriers and AMLs must return to the same spot from which they start their tour (close routes), and each courier is assigned to only one route (single trip). To solve their problem in large instances, they develop an extended version of the Clarke and Wright heuristic (ECWH). We call the problem defined in Jun Li et al. (2021) a two-echelon vehicle routing problem with simultaneous pickup and delivery, abbreviated by 2E-VRP.

In the current work, AMLs and couriers can both perform open routes which leads to cost savings because it relaxes the constraint forcing AMLs and couriers to complete their mission at the same spot from which they started. Moreover, we assume that couriers can perform multiple missions, which allows couriers to start a new mission at any stop. We hypothesise that 2E-MOVRP will lead to a better solution in terms of the total cost compared to 2E-VRP. To better investigate the cost-saving associated with considering the open route operations, we modified the algorithm developed by Jun Li et al. (2021) to assign the final closed routes to the minimum number of couriers, observing the maximum length for a courier mission. We call this version 2E-MVRP, which we deploy ECWH to solve. We also compare the results of 2E-MOVRP with the courier-only problem where couriers perform pickup and delivery operations in a single-echelon network. Note that, for the courier-only problem, the couriers are allowed to perform multiple missions. The notation used in this section to show the results are summarised in Table 2.2.

Table 2.2. The abbreviations used for comparative analysis

Acronym	Description	Algorithm
2E-MOVRP	Two-echelon multi-mission pickup and delivery routing problem	ABSM
2E-MVRP	Two-echelon multi-mission pickup and delivery routing problem	ECWH
2E-VRP	Two-echelon pickup and delivery routing problem	ECWH

To test our heuristic, networks ranging from 50 to 150 customers in grid dimensions of 1 km^2 to 36 km^2 are randomly generated following the process carried out by Jun Li et al. (2021) for instance generation. For the networks with 50, 100, and 150 customers, we considered 2, 3, and 5 depots, respectively. In order to solve 2E-MOVRP, the ABSM algorithm is applied 10 times to each instance, the best solution of which is reported for the comparative analysis. Then, for a given network configuration, the results corresponding to 2E-MOVRP, 2E-MVRP, 2E-VRP, and courier-only algorithms are obtained for four different random datasets. Figures 2.6 (a) to 2.6 (d) provide a schematic comparison of these algorithms for a network of 50 customers randomly generated in an area of 4×4 km^2. In Figure 2.6, routes of the same colour are performed by the same courier.

Figure 2.7 shows the objective values calculated from the implementation of algorithms for samples with 50, 100 and 150 customers in networks with dimensions of 1 km^2, 4 km^2, 16 km^2 and 36 km^2.

Among all these models, 2E-MOVRP and 2E-VRP (couriers can only perform single missions) offer the lowest cost and highest cost, respectively. The reason is that more couriers are required, leading to an increase in fixed cost. The results also show that the cost in a two-echelon network of couriers and AMLs is less than when only couriers making multiple missions are employed. Another noteworthy point is that the objective value increases for larger networks with greater slope.

Therefore, it is expected to have larger cost savings for larger networks. In order to compare these algorithms in pairs, we calculated the cost savings as a percentage, as shown in Figure 2.8. Figure 2.8 shows that the savings obtained by comparing 2E-MOVRP with 2E-VRP, 2E-MVRP, and courier-only can be achieved up to approximately 85%, 72%, and 83%, respectively, for a network with 1 km^2 area.

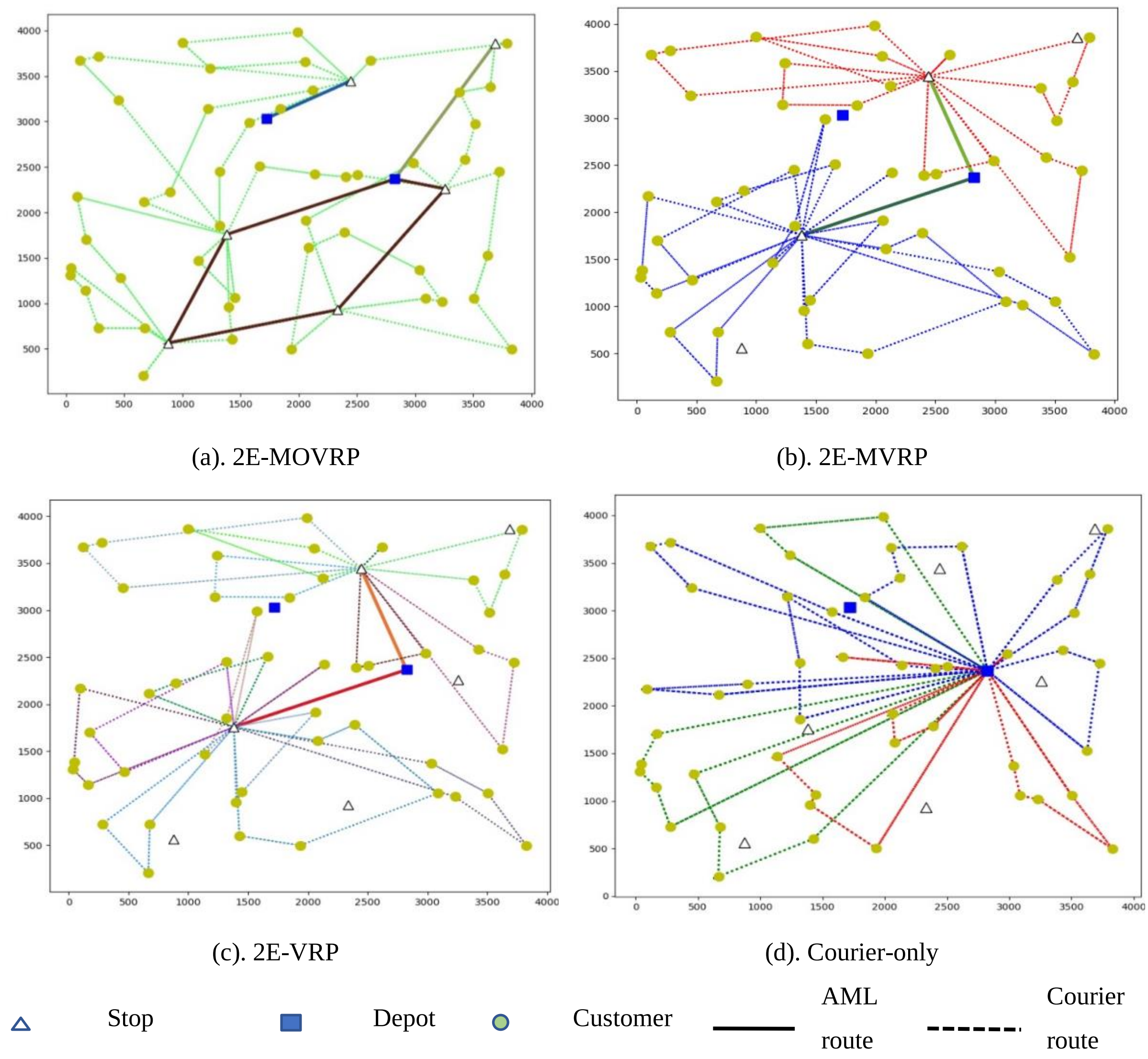


Figure 2.6. Schematic review of 2E-MOVRP, 2E-MVRP, 2E-VRP, and Courier-only

The greatest cost-saving is seen in networks with high customer density. As the customer density increases, so does the cost saving. Also, the greatest cost saving is obtained by comparing 2E-MOVRP with 2E-VRP.

Although the courier-only algorithm operates in a single-echelon non-cooperative system, its objective results in a lower cost than the 2E-VRP because of the possibility of multiple missions being executed by couriers (in 2E-VRP addressed in this chapter, couriers are not allowed to perform multiple tours). This significantly reduces the number of couriers required for pickup and delivery missions. Also, if only couriers are used to serve the customers, it significantly increases the value of the objective. The comparison between courier-only and 2E-MOVRP shows that cost-savings of approximately 83% and 43% can be achieved in the networks with the highest and lowest customer density, respectively. Figure 2.9 illustrates the CPU time for 2E-MOVRP, 2E-MVRP, 2E-VRP, and courier-only algorithms.

To show the efficiency of our algorithm in terms of the execution time, we generate 4 random instances for each network configuration with different number of customers and dimensions. For instance, 50 customer nodes are randomly generated 4 times in network dimensions of 1×1 km^2, 2×2 km^2, 4×4 km^2 and 6×6 km^2. The same procedure is applied for 100 and 150 customers. In total, each algorithm is examined under 48 numerical experiments.

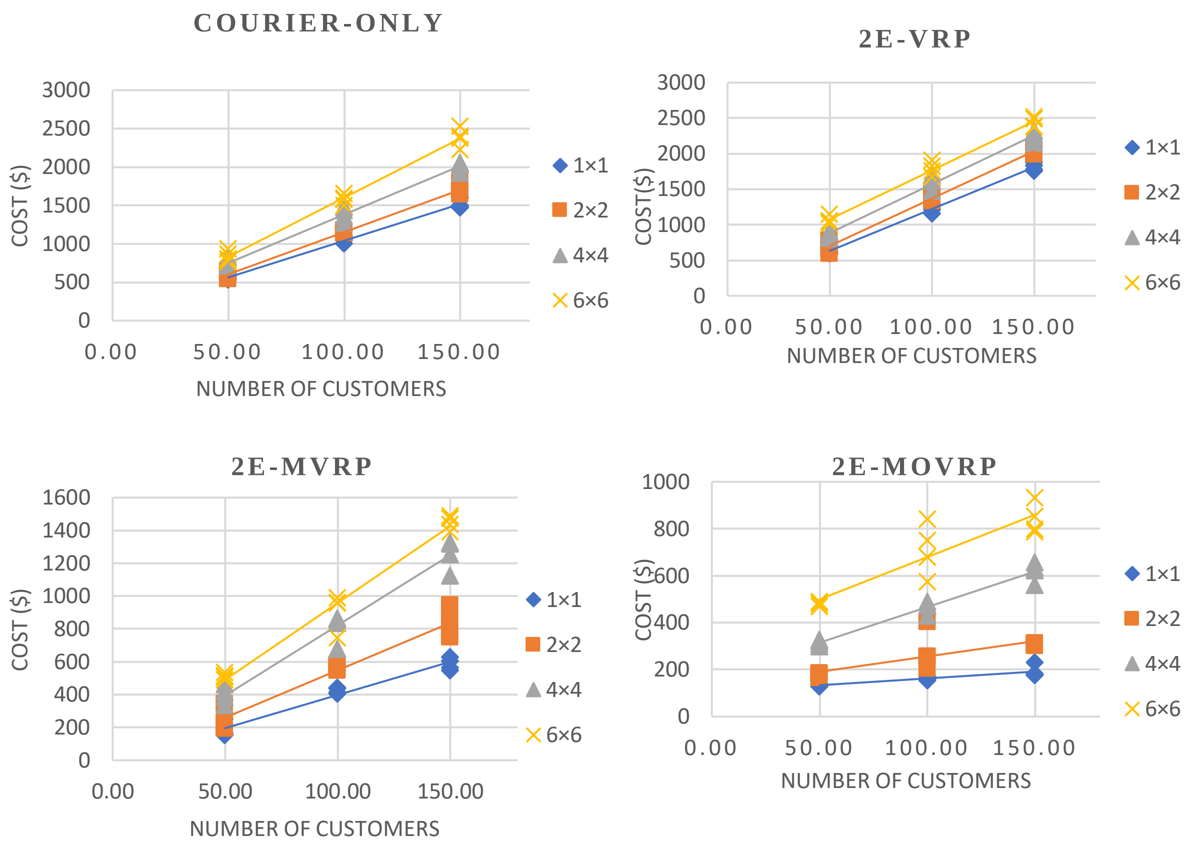


Figure 2.7. The objective value of 2E-MOVRP, 2E-MVRP,2E-VRP, Courier-only algorithms in the networks with 50,100,150 customers

As previously mentioned, we apply ABSM algorithm described in Section 2.5 to solve 2E-MOVRP, while ECWH introduced by Jun Li et al. (2021) is used to solve 2E-MVRP, 2E-VRP, and Courier-only models. As seen in Figure 2.9, our proposed algorithm for 2E-MOVRP, even with larger solution space, can solve the instances with a considerably lower execution time compared with 2E-MVRP and 2E-VRP. The average solution time for ABSM algorithm is approximately 22, 257, and 330 seconds for 50, 100, and 150 customers, respectively. For 2E-MVRP, these values are 57, 180, and 2235 seconds.

Although comparable for instances with 50 and 100 customers, the two algorithms perform quite differently for large sizes in terms of the CPU time. As seen in Figure 2.9, with an increase in the number of customers from 100 to 150, a sharp increase is seen in the CPU time of the ECWH algorithm. ABSM is more stable against an increase in the size of the networks. As observed, an increase of an average of 73 seconds is visible in return for increasing the customer nodes from 100 to 150, whereas this increase is 2055 seconds and 1830 seconds for 2E-MVRP and 2E-VRP, respectively.

2.7 SENSITIVITY ANALYSIS

In this section, the impact of changing operational parameters—AML carrying capacity and range—on the objective value is investigated. The purpose of this analysis is to provide insights for decision-makers in system design.

To implement the sensitivity analysis, a network of 2×2 km^2 with 50 customers, a network of 4×4 km^2 with 100 customers, and a network of 6×6 km^2 with 150 customers are randomly generated. Then, for each instance, four different ranges of AML capacity and maximum distance range are considered. Figure 2.10 shows the impact of these parameters on the total cost. The results show that even for the largest network in this study with the largest number of customers, a maximum AML range of more than 3 km will not further reduce total cost. However, increasing AML capacity decreases the cost. For instance, for a network with a size of 6×6 km^2 and 150 customers, increasing the capacity from 100kg to 700kg reduces the total cost by 30.5%. Also, by increasing the range of AMLs from 1 to 3 km, the objective value is improved by 16.8% in the network with a size of 6×6 km^2 and 150 customers. Based on the results presented in this section, it can be concluded that the focus on increasing AML capacity has a more significant contribution in reducing costs than increasing the range of AMLs.

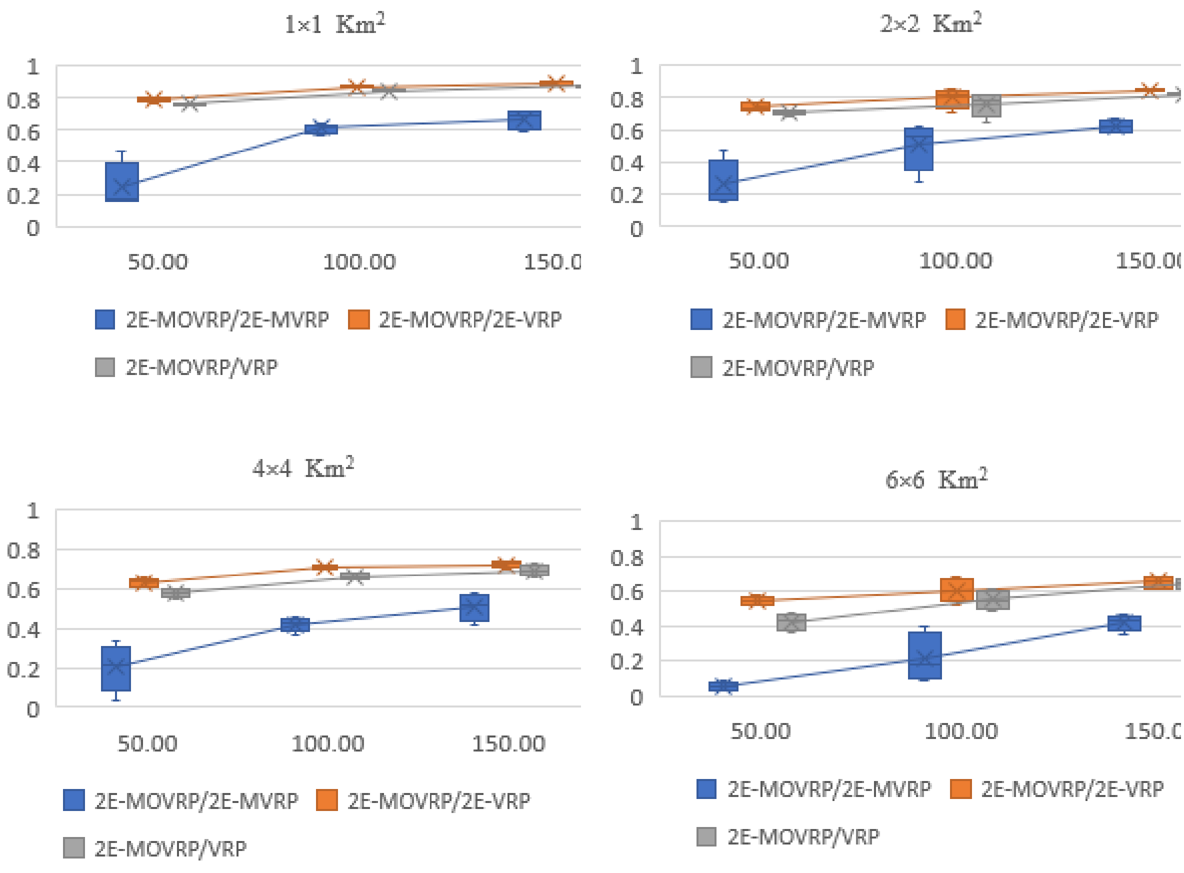


Figure 2.8. The cost savings in each grid dimension

2.8 CONCLUSION

This study examined a collaborative human-robot system for a simultaneous delivery pickup problem. In this model, AMLs supply couriers with deliveries and collect pickups from couriers. Only couriers serve the end customers, while only AMLs access the depots. In order to model this problem, a novel MILP formulation was presented. The proposed MILP model incorporates a two-echelon network of multiple depots, multiple stops, *m*-AMLs, and *n*-couriers. In this model, each courier is allowed to perform multiple open routes, enabling couriers to start a new mission from any stop, subject to constraints relating to payload and total distance. The objective function is to minimize the total fixed and variable costs, whereby fixed cost relates to the number of AMLs and couriers deployed and variable cost relates to the total distance covered by AMLs and couriers.

Since the proposed MILP model is not able to solve large-scale problems, an algorithm combining a number of heuristics was developed whose performance was tested using several numerical experiments. The results show that the algorithm is able to provide the optimal solutions for most small-size instances. We also designed an experiment to further examine the efficiency of the algorithm. For this purpose, the solution obtained from solving the algorithm was given to the MILP model as an initial solution. The results showed that after 1800 seconds, the MILP model can rarely improve the initial solution for medium-sized networks.

Also, the proposed algorithm for solving the large-scale instances was compared with three other variants of the simultaneous pickup and delivery problem. Since this study is an extension of the study by Jun Li et al. (2021), the results of both studies were compared in terms of both objective function and solution time. The results show that significant cost savings be achieved by using a robot-human cooperative system where couriers are allowed to perform multiple open routes.

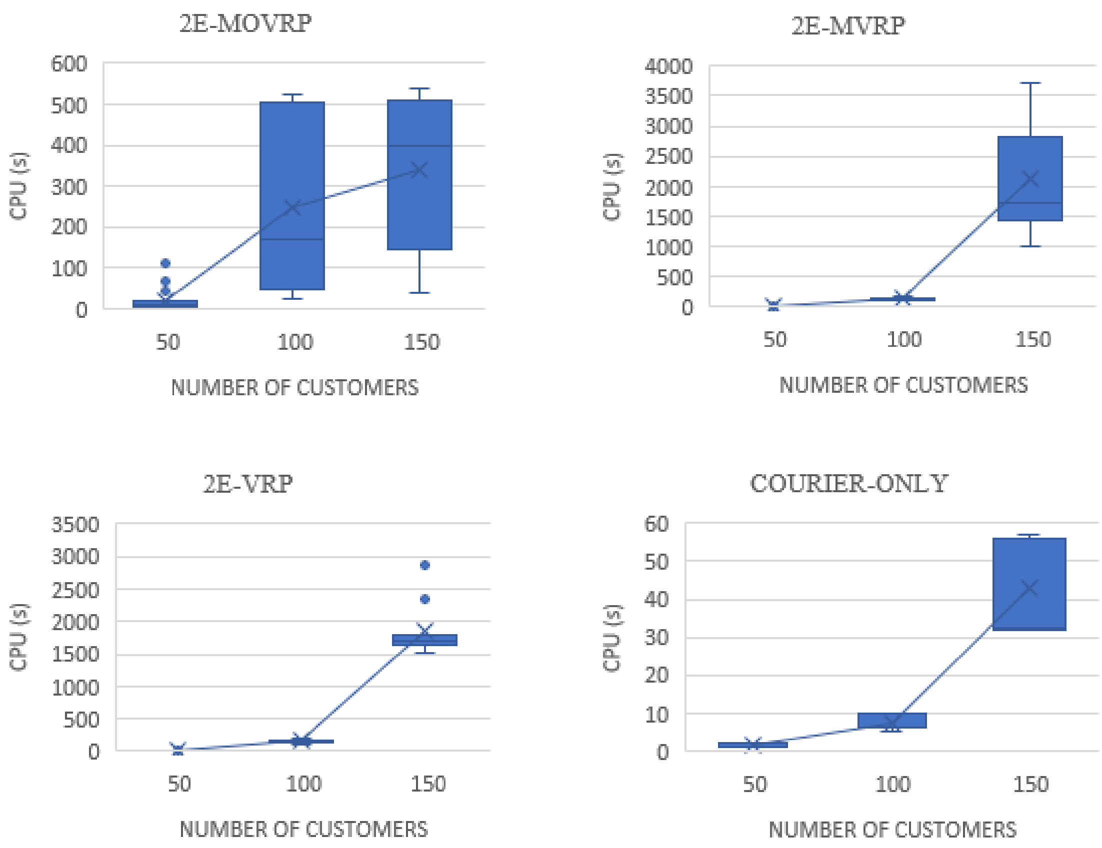


Figure 2.9. The CPU time of the ABSM and ECWH algorithms for solving 2E-MOVRP, 2E-MVRP, 2E-VRP, and Courier-only

Finally, the impact of the capacity and maximum range of the AMLs on the total cost was analysed. The results show that increasing the capacity from 300 to 700 kg in a network with a size of 6×6 km^2 and 150 customers can reduce the total cost by up to 30%.

Figure 2.10. Sensitivity analysis on the AML capacity (left) and AML maximum range

Also, for the same network configuration, a 16.8% improvement in the total cost was reported by increasing the range of AMLs from 1 to 3 km. Based on the sensitivity analysis, it was concluded that focusing on the use of AMLs with higher capacity has a greater effect on reducing costs compared to increasing their maximum range.

Future studies can incorporate scheduling considerations into modelling. The current study also does not consider energy consumption equations, which future studies could provide a more real-world routing model focusing on physical characteristics and environmental factors involved in AML energy consumption. The development of solution algorithms that are able to provide quality solutions in a shorter CPU time is always welcomed.

Having established a comprehensive framework for optimizing pickup and delivery operations with the introduction of the hybrid Autonomous Mobile Locker (AML) and courier delivery platform in Chapter 2, we now shift our focus towards further enhancing the efficiency and flexibility of delivery systems. In Chapter 3, we introduce a novel approach by integrating drones into the conventional logistics network, paving the way for a flexible Truck-Drone Delivery System. This innovative integration aims to capitalize on the strengths of both aerial and ground-based delivery methods, ultimately offering a solution to address the evolving demands of modern delivery operations.

Appendix 2.A: Pseudo codes

Algorithm 2.1: Backtracking

Input:

$CloseNeighbors$ ←for each node, sort other nodes based on their distance in the ascending order.
$MaxNeighbor$ ← calculate the number of maximum number of neighbors which BT should branch based on the **adaptive-*k*-nearest-neighbour** procedure.
$MaxNeighbor(i)$ ←for each stop consider $StopK$ as the maximum number of neighbours used in BT.

Initialize:

Vs ← {False, $\forall i \in N_C \cup \{Current, Dest\}$}

$Prev$ ←the origin, $Neighbor \leftarrow 1$

$Current$ ←the origin, $Dest$ ← the destination, $path \leftarrow \emptyset$, $AllPath \leftarrow \emptyset$,

GeneratePath$(Current, Dest, Vs, Prev, path)$:
    if (the number of nodes in $path \leq 2$ & $Current = Dest$**) then**
        $Vs(Current)$ ←False
    else
        $Vs(Current)$ ←True
    end If
    $path \leftarrow path \cup \{Current\}$
    if (the number of nodes in $path > 2$ & $Current = Dest$ & $path$ is feasible**) then**
        $AllPath \leftarrow AllPath \cup path$
    end If
    for (each node $Next \in CloseNeighbors(current)$ **) do**
        if ($Vs(Next) = False$ &$Neighbor \leq MaxNeighbor(Current)$ & $path \cup \{Next\}$ is feasible & $path$ satisfies load-to length-ratio**) then**
            GeneratePath$(Current, Dest, Vs, Prev, path)$

$Neighbor \leftarrow Neighbor + 1$

end If

end for

end GeneratePath

Output: $AllPath$

Algorithm 2.2: Simulated annealing (SA)

Input:

$nMove, Temp, ET, DistRate, MaxNoImprove$,$CooolingRate$,$MinTemp, MinMove, \alpha, \beta$

Initialize:

$CurrentSol, BestSol \leftarrow$ Courier routes, generated by BT

$Continue \leftarrow$ True

$NoImprove \leftarrow 0$

SA($CurrentSol$**)**:

While ($Continue$**) do**

$NewSol \leftarrow$**MoveToNeighbor**($CurrentSol$, $nMove$, $DistRate$)

if (Total cost of $NewSol$ < Total cost of $BestSol$**) then**

$CurrentSol \leftarrow NewSol$

$BestSol \leftarrow NewSol$

else do

if (Total cost of $NewSol$ < Total cost of $CurrentSol$**) then**

$CurrentSol \leftarrow NewSol$

else if (Metropolis criteria accepts $NewSol$**) then**

$CurrentSol \leftarrow NewSol$

end if

end if

if (for the last ET iterations $BestSol$ is not improved**) then**

$Temp \leftarrow Temp \times CooolingRate$

$nMove \leftarrow (1 + \beta) \times nMove$

$DistRate \leftarrow min\{DistRate \times (1 + \alpha), 1\}$

$NoImprove \leftarrow NoImprove + 1$

else do

$NoImprove \leftarrow 0$

$nMove \leftarrow min\{(1 - \beta) \times nMove, MinMove\}$

$DistRate \leftarrow DistRate \times (1 - \alpha)$

end if

If ($NoImprove > MaxNoImprove$ **or** $Temp < MinTemp$**) then**

$Continue \leftarrow$False

26: **end if**
27: **end while**
28: **end** SA
Output: $AllPath$

Algorithm 2.3: Merging Routes

Inputs:
$dist, rc, N_V, CurentSol$
Initialize:
01: $\mathfrak{R}^{best}$, $\mathfrak{R}$ ←the courier routes in $CurentSol$
02: $r^{merged} \leftarrow \emptyset$
03: the courier routes in $CurentSol$
04: **for (**$r \in \mathfrak{R}$**) do**
05: $\mathfrak{R}^{new} \leftarrow \emptyset$
06: $\mathfrak{R}' \leftarrow r$
07: $r^{merged} \leftarrow r, \mathfrak{R}'/\{r\}$
08: **while (**$\mathfrak{R}' \neq \emptyset$**) do**
09: r^{next} ← A route in $\mathfrak{R}'$, which has either the closest origin or destination to the destination of r^{merged}.
10: **if (**the destination of r^{next} is closer to the destination of r^{merged}**) then**
11: **if (**the reverse of r^{next} is feasible**) then**
12: r^{next} ← reverse of r^{next}
13: **end if**
14: **end if**
15: L^C ← the last costumer visited in r^{merged}
16: L^S ← the last stop visited in r^{merged}
17: F^C ← the first costumer visited in r^{next}
18: F^S ← the first stop visited in r^{next}
19: **if (**$dist_{L^C F^S} + dist_{F^C F^S} < dist_{L^C L^S} + dist_{L^S F^S}$**) then**
20: $r^{merged} \leftarrow r^{merged}/\{L^S\}$
21: **else do**
22: $r^{next} \leftarrow r^{next}/\{F^S\}$
23: **end if**
24: $r^{merged} \leftarrow r^{merged} \cup r^{next}$
25: **if (**the length of $r^{merged} > rc$**) then**
26: $r^{merged} \leftarrow r^{merged}/r^{next}$
27: $\mathfrak{R}^{new} \leftarrow \mathfrak{R}^{new} \cup r^{merged}$
28: $r^{merged} \leftarrow r^{merged} \cup \{F^S \cup r^{next}\}$
29: **end if**
30: **end while**
31: $\mathfrak{R}^{new} \leftarrow \mathfrak{R}^{new} \cup r^{merged}$
32: $Continue$ ←True
33: **for each (**$r \in \mathfrak{R}^{new}$**) do**
34: r ← replace the last stop of r with its first stop
35: Update r in $\mathfrak{R}^{new}$
36: **end for**
37: **if (**the length of $\mathfrak{R}^{new} \leq rc$ & the cost of $\mathfrak{R}^{new} \leq$ the cost of $\mathfrak{R}^{best}$ **) then**
38: $\mathfrak{R}^{best} \leftarrow \mathfrak{R}^{new}$
39: **end if**
40: **end for**
41: **for (**$r \in \mathfrak{R}$**) do**
42: **if (**Reverse of r is feasible**) then**
43: Repeat (4-41) for the reverse of r
44: **end if**
46: **end for**

Output: $\mathfrak{R}^{best}$

Appendix 2.B. Computational experiment on algorithm parameters

The algorithm presented in this paper uses eight parameters, all of which have significant impact on the solution process, computation time, and the quality of the solution. In order to find good values for these parameters, an experimental design method is used. For each parameter, different values are set. These values are determined by the experience gained from a large number of numerical experiments while testing the algorithm in various settings. The values for each of the parameters are given in Table 2.3.

Table 2.3. The values of each parameter used in the algorithm

Parameter	Levels
$nMove$	{2, 3, 4, 5} $\times \vee N \vee$
$Temp$	{5, 10, 20, 100}
$EquibliriumTenure$	{5, 10, 15, 20, 40}
$DistRate$	{0.1, 0.3, 0.5, 0.7}
$LoadToLenghtRatio$	{0.5, 0.83, 1.25,1.43}
$Max_{NoImprove}$	{2, 3, 4, 5}
k	{3, 4, 5, 6}
$StationK$	{0.05, 0.1, 0.2, .25, .50}$\times \vee N \vee$

The values of $nMove$ and $StationK$ are set according to network size. This prevents unnecessary search processes for small networks while a wider search space for larger networks is examined. This approach also reduces the probability of getting stuck in local optima during the early iterations of running the algorithm for large-size instances.

The number of total combinations of the parameter values is equal to 102400, which makes it virtually impossible to test all these combinations for networks of different sizes. Therefore, for networks with the size of 10 to 30 nodes, 1000 different configurations including a combination of parameter values are randomly generated. After performing the tests, we use a measure called Dominance Index to check the performance of the configuration for each data set. To quantify this index, we compare the performance of these configurations in pairs in terms of the objective function and the execution time. This value indicates how many other configurations a particular configuration outperforms in terms of both the objective function and the run time. Finally, the dominance values of each configuration are added together for different datasets. To select the base configuration, 20 configurations with the highest dominance values are selected. Table 2.4 shows the parameter values for the 20 dominant configurations.

Table 2.4. The parameter values for the 20 dominant configurations

Configuration	$nMove$	$Temp$	DR	ET		$NoIm$	k	SK
1	5	100	0.5	10	0.83	5	5	0.10
2	5	20	0.3	15	1.43	5	4	0.10
3	4	10	0.1	40	0.83	4	5	0.05
4	5	20	0.1	20	1.25	3	6	0.50
5	4	10	0.7	15	1.25	5	4	0.25
6	4	100	0.1	40	0.83	3	4	0.25
7	4	5	0.3	20	1.43	3	4	0.25
8	4	10	0.7	5	1.43	3	4	0.50
9	4	10	0.7	5	0.50	3	5	0.20
10	3	5	0.1	20	1.43	2	5	0.20
11	4	10	0.7	15	0.83	4	5	0.50
12	5	20	0.5	15	0.83	4	5	0.05
13	3	10	0.5	10	0.83	5	3	0.25
14	5	100	0.3	20	1.43	4	5	0.10
15	2	100	0.1	20	0.50	5	4	0.20
16	3	20	0.7	20	1.25	2	3	0.20
17	4	10	0.7	5	1.25	4	6	0.50
18	3	10	0.3	40	1.43	4	3	0.05
19	5	5	0.1	15	1.43	2	4	0.10
20	4	5	0.7	15	1.25	3	4	0.05

In the continuation of the experiments, instances of networks with 40, 60, 80, and 100 nodes with an area of 4×4 square kilometres are considered. For each grid size, 3 instances are generated randomly. Thus, there will be 4 × 3 = 12 datasets for each configuration. Each experiment is repeated 10 times for each configuration and the best solution in terms of the objective function is reported. Once again, we use the dominance index to check the performance of each of the 20 configurations for all data sets. Table 2.8 shows the dominance value of each configuration for different datasets.

Table 2.5. The dominance value of each configuration

configuration	N=40	N=60	N=80	N=100	Sum
1	3	0	0	1	4
2	2	9	8	2	21

3	0	0	1	0	1
4	3	1	0	15	19
5	3	9	1	1	14
6	1	2	0	9	12
7	9	3	4	0	16
8	4	9	0	0	13
9	0	3	3	6	12
10	0	1	2	9	12
11	0	1	2	4	7
12	2	0	1	0	3
13	1	5	5	16	27
14	2	4	2	0	8
15	0	2	0	11	13
16	6	4	6	18	34
17	7	8	0	5	20
18	1	5	4	17	27
19	2	2	4	9	17
20	5	5	1	0	11

According to Table 2.8, Configuration (16) with a total score of 34 for the dominance index is selected as the basic configuration. Figure 2.11 shows the overall performance of all configurations in terms of the objective function and the average solution time. Configuration (16) is marked with a square shape. In each dataset, configurations whose value of the objective function and solution time are closer to the origin of the coordinates perform better.

In order to more accurately investigate the effect of parameters on the solving mechanism of the algorithm and on improving the objective function, we perform a sensitivity analysis on these parameters. During this analysis, we set all the parameters equal to the corresponding value in the base configuration (Config. 16) and test each parameter based on the values introduced in Table 2.5. Each of the tests includes instances of grids with sizes of 40, 60 and 100 nodes in an area of 4×4 km^2.

For example, Figure 2.12 shows the effect of different $DistRate$ values on the objective function and the execution time. The value of $DistRate$ in the base configuration is 0.7. Among the parameters of the algorithm, $DistRate$ has the greatest impact on the improvement of the objective function. According to Figure 2.12, it can be seen that the higher the $DistRate$ value, the greater the possibility of achieving a solution with a better objective function.

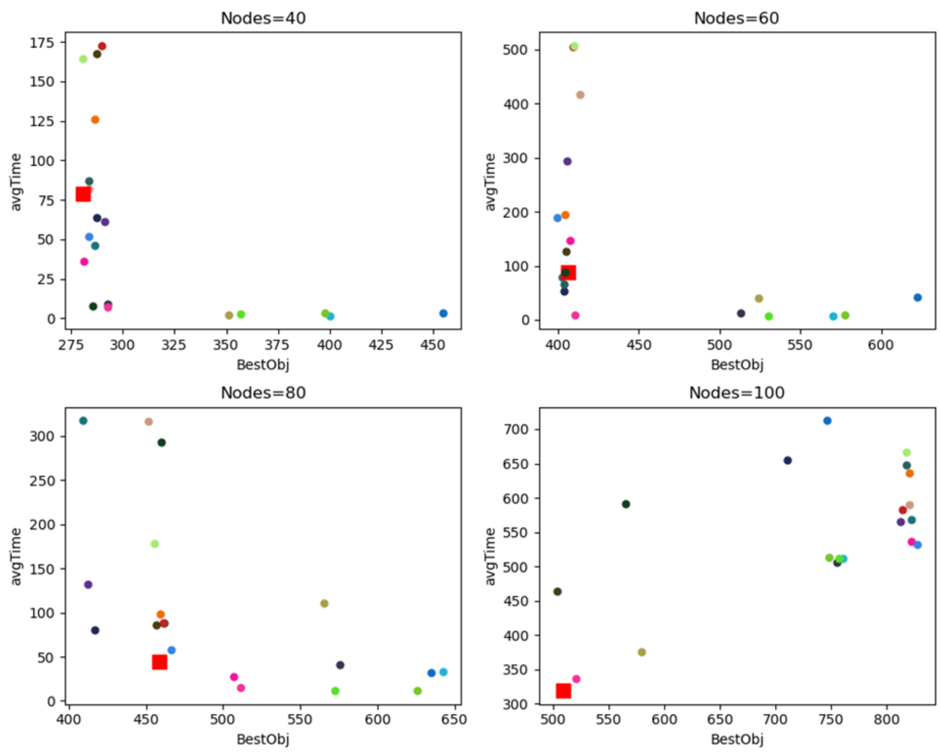


Figure 2.11. The schematic view of 20 configurations in different problem sizes

The vertical axis on the left represents the solution time of the algorithm and the vertical axis on the right represents the best objective function. Red triangles and black squares show the solution time and the objective function value obtained according to different $DistRate$ values, respectively. Therefore, our recommended value for the $DistRate$ parameter is 0.7.

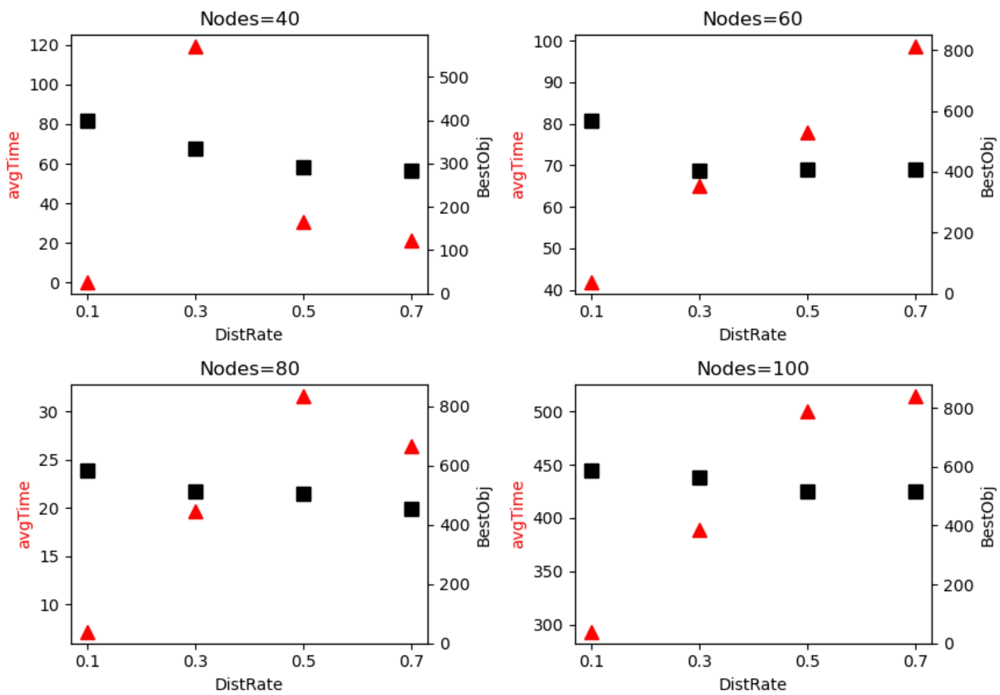


Figure 2.12. Sensitivity analysis on different $DistRate$ values with instances of grids with sizes of 40, 60 and 100 nodes in an area of 4 × 4 square kilometres.

Similarly, other parameters are tested to determine their best values. Table 6 shows the best values obtained for the algorithm parameters after performing the sensitivity analysis tests.

Table 2.6. The best values of each parameter used in the algorithm

Parameter	Best value
$nMove$	$3 \times \lvert N \rvert$
$Temp$	20
$EquibliriumTenure$	10
$DistRate$	0.7
$LoadToLenghtRatio$	1.25
$Max_{NoImprove}$	2
k	3
$StationK$	$0.2\times \lvert N \rvert$

Appendix C: Numerical results

Table 2.7. Performance comparison between the MIP solver and the algorithm for small instances

	$(\lvert N_D \rvert, \lvert N_S \rvert, \lvert N_C \rvert)$= (1,1,10), |ω|=1 km^2						$(\lvert N_D \rvert, \lvert N_S \rvert, \lvert N_C \rvert)$= (1,2,15), |ω|=1 km^2						$(\lvert N_D \rvert, \lvert N_S \rvert, \lvert N_C \rvert)$= (2,3,20), |ω|=1 km^2					
Inst	Gurobi		Heuristic			Inst	Gurobi		Heuristic			Inst	Gurobi			Heuristic		
	$Best$	sec	$Best$	std	dev (%)		$Best$	sec	$Best$	std	dev (%)		$Best$	LB	sec	$Best$	std	dev (%)
A1	110.50	3.94	110.50	10.47	0	B1	103.61	242	103.61	10.75	0.00	C1	119.86	110.59	1800	111.96	8.72	-6.59
A2	105.59	1.12	105.59	4.74	0	B2	103.61	1660	103.61	9.44	0.00	C2	115.83	112.45	1800	115.83	9.61	0.00
A3	110.12	5.77	110.12	3.83	0	B3	109.11	612	109.19	9.42	0.07	C3	113.86	110.23	1800	116.45	10.01	2.27
A4	112.00	5.48	112.00	8.87	0	B4	109.34	820	109.34	8.98	0.00	C4	113.02	112.87	1800	113.02	11.74	0.00
A5	110.68	2.32	110.68	2.70	0	B5	112.33	1235	113.24	6.00	0.80	C5	113.27	109.43	1800	111.47	8.18	-1.58
A6	111.76	1.69	111.76	2.19	0	B6	112.74	940	112.89	11.87	0.10	C6	119.30	114.17	1800	116.42	8.13	-2.41
A7	108.39	7.95	108.39	3.09	0	B7	114.35	620	114.35	7.56	0.00	C7	113.97	113.97	1615	113.97	8.54	0.00
A8	104.41	9.45	104.41	9.30	0	B8	111.00	580	111.00	11.48	0.00	C8	124.46	117.12	1800	119.39	11.98	-0.40
A9	106.47	3.32	106.47	7.92	0	B9	113.86	113	114.39	9.86	0.40	C9	115.52	113.67	1800	115.52	6.09	0.00
A10	115.75	3.87	115.75	2.43	0	B10	110.06	870	110.06	12.55	0.00	C10	114.45	114.45	1718	116.99	10.69	-2.20
A11	113.75	7.06	113.75	9.47	0	B11	113.19	630	113.19	7.04	0.00	C11	111.26	110.79	1800	111.26	11.37	0.00
A12	110.30	1.28	110.30	10.01	0	B12	111.40	740	111.40	5.83	0.00	C12	113.27	109.35	1800	115.29	8.59	1.78
A13	105.90	5.26	105.90	8.09	0	B13	108.35	1440	110.97	9.37	2.40	C13	116.17	113.56	1800	114.66	9.96	-1.75
A14	106.10	1.64	106.10	6.97	0	B14	107.43	370	107.43	9.77	0.00	C14	112.65	112.65	1517	113.03	12.63	0.33
A15	110.08	1.93	110.08	8.96	0	B15	110.75	420	110.75	5.95	0.00	C15	117.39	112.66	1800	114.16	10.95	-2.75
A16	110.63	3.92	110.63	7.02	0	B16	109.42	1125	111.80	5.17	2.10	C16	117.89	116.35	1800	117.65	12.29	-0.20
A17	107.41	4.15	107.41	4.57	0	B17	113.77	268	113.77	10.12	0.00	C17	112.19	112.19	1612	114.78	9.40	2.30
A18	107.18	1.35	107.18	4.26	0	B18	110.23	339	110.23	5.93	0.00	C18	116.55	112.74	1800	115.11	10.66	-1.23
A19	110.29	9.61	110.29	5.28	0	B19	108.74	927	111.09	10.37	2.90	C19	115.43	113.89	1800	115.88	10.35	0.39
A20	105.95	4.14	105.95	4.94	0	B20	108.65	537	108.65	10.57	0.00	C20	114.79	112.34	1800	115.67	10.88	0.76
Mean	109.16	4.26	109.16	6.25	0		110.09	724	110.58	8.91	0.44		115.55	112.77	1763.1	114.92	8.72	-0.56

Table 2.8. Performance comparison between the MIP solver and the algorithm for medium instances

	Heuristic			Gurobi			
Ins.	$Best$	std	sec	UB	LB	sec	imp (%)
D1	125.43	9.41	3	125.34	121.72	1800	0.07
D2	121.51	5.53	5	121.51	117.66	1800	0
D3	123.89	2.88	8	123.89	120.36	1800	0
D4	128.81	3.00	2	128.81	125.32	1800	0
D5	124.57	3.96	1	124.57	119.74	1800	0
D6	134.58	5.16	5	134.58	128.02	1800	0
D7	128.17	5.94	6	128.17	123.76	1800	0
D8	121.26	10.04	4	121.26	115.07	1800	0
D9	130.29	5.92	2	130.29	122.70	1800	0
D10	128.89	7.55	9	128.89	105.43	1800	0
Mean	126.74	5.94	4.5	126.73	119.97	1800	0.007

Chapter 3: A Hybrid Vehicle-drone Delivery Routing Problem

3.1 INTRODUCTION

Over the past decade, the advent of new technologies and the pervasiveness of the internet has led to significant transformations within the retail sector. While traditional brick-and-mortar retailers have moved towards adopting an omnichannel strategy, integrating physical and online sales channels, pure-play e-retailers have also experienced significant growth (Ailawadi & Farris, 2017; Thakur, 2018). These developments have shifted the focus of competition from product ranges and pricing to the quality of order fulfilment services (Zhang et al., 2019). Last-mile logistics planning continues to present challenges for retailers (Bhattacharjya et al., 2016; Hübner et al., 2016; Ishfaq et al., 2016).

Transportation accounts for the majority of last-mile delivery costs, which have been traditionally carried out by ground vehicles such as trucks and vans. The use of traditional ground vehicles, however, results in a significant waste of time due to traffic congestion, searching for a parking space and making the ultimate delivery on foot. Unmanned aerial vehicles (UAVs), colloquially known as drones, have recently received attention as potential package delivery agents (Valavanis & Vachtsevanos, 2015). Drones have significant advantages over conventional vehicles, including the potential to ship goods at lower operating costs and faster, because they perform the delivery task autonomously and do not face complex obstacle avoidance scenarios or traffic congestion (Koster, 2015).

While a drone is fast with relatively inexpensive operational costs, there are some drawbacks that limit its use. The most obvious limitation of drones is their inability to carry heavy or large packages (Shi et al., 2018). This means that drones, unlike trucks, cannot be used to deliver boxes to a large number of customers on one tour. Another limitation of drones is their battery capacity which limits their flight duration and therefore their range. Given these limitations, it can be argued that using a fleet of drones to deliver shipments to customers may not be a practical solution on its own. However, cooperation between a fleet of drones and trucks can solve many of these limitations (Murray & Chu, 2015).

In the collaborative truck-drone system, the drones pick up packages from the truck and deliver them to customers without returning to the depot after each mission (Dorling et al., 2016). A depleted drone battery can also be swapped by a truck operator with a fully charged battery, allowing the drone to carry out more than one delivery tour. Both the truck and the drone deliver the shipments to the customers in such a way that customers' demands are assigned to each vehicle type based on their location and the features of the orders.

This study develops a novel mixed-integer programming (MIP) model for a cooperative system of multiple drones and trucks where drones can land on a truck that is not necessarily its 'mother truck' (or truck-of-origin) and then be dispatched on another mission. This assumption allows the drones to potentially use more of their flight range for making deliveries, making the system more flexible and resulting in considerable cost savings. Dispatching drones takes place at customer locations and docking stations, while trucks can retrieve drones only at docking stations. Both trucks and drones deliver packages to customer in a synergistic way. Routes and the assignment of consignments to trucks and drones are jointly optimized by taking into account the size and the weight of packages as well as customer locations.

3.1.1 Contributions

The main contributions of this study are as follows:

- Flexible Drone-Truck Retrievals: The system allows drones to land on any available truck for package retrieval, offering flexibility in the retrieval process. This flexibility optimizes resource utilization within the network.
- Efficient Multi-Trip Routing: Our framework supports the dispatch of drones multiple times from the same truck stop location to consecutively deliver items to multiple customers. This concept embodies the complexity of the Multi-trip Vehicle Routing Problem with Drones, providing efficient solutions.
- Multi-Visit Deliveries by Drones: Drones are empowered to make multiple deliveries per mission, addressing the challenges posed by the Multi-visit Vehicle Routing

Problem with Drones. This feature enhances the overall efficiency of the delivery system.

- Diverse Truck Strategies: Within the system, trucks have the option to employ both the dispatch-wait-collect and dispatch-move-collect strategies, offering adaptability in response to varying operational demands.
- Innovative Heuristic Approaches: This research introduces two novel heuristics based on the extended Clarke and Wright algorithm. These heuristics are designed to efficiently solve large-scale problem instances, demonstrating their capacity to optimize complex delivery scenarios.

The remainder of this chapter is organized as follows. In Section 3.2 a review of related literature is presented. Section 3.3 rigorously defines the research problem. In Section 3.4 the mathematical formulation is presented. Section 3.5 proposes a solution methodology. In Section 3.6, results, and sensitivity analyses for a range of problem sizes are presented and discussed. Finally, the conclusion is provided in Section 3.7.

3.2 REVIEW OF THE RELEVANT WORKS

This section reviews the related literature and provides a comparative analysis with previous works. Then, we identify the research gap and present an overview of the main differences between our work and the existing studies.

Drones are an emerging technology whose usage in day-to-day operations is becoming more prevalent. Optimization problems pertaining to trucks and drones have gained increasing interest among researchers over the past years. Otto et al. (2018) have categorized more than 300 papers into four main groups:

- Vehicles supporting drone operations.
- Drones supporting vehicle operations.
- Drones and vehicles executing separate tasks.
- Drones and vehicles functioning as coordinated units.

The drone-truck routing problem falls into two major categories: the traveling salesman problem with drones (TSP-D) and the vehicle routing problem with drones (VRP-D). In addition, two more micro subcategories may be derived from TSP-D; the Flying Sidekick TSP (FSTSP) and the parallel drone scheduling TSP (PDSTSP). Figure 3.1 illustrates this classification.

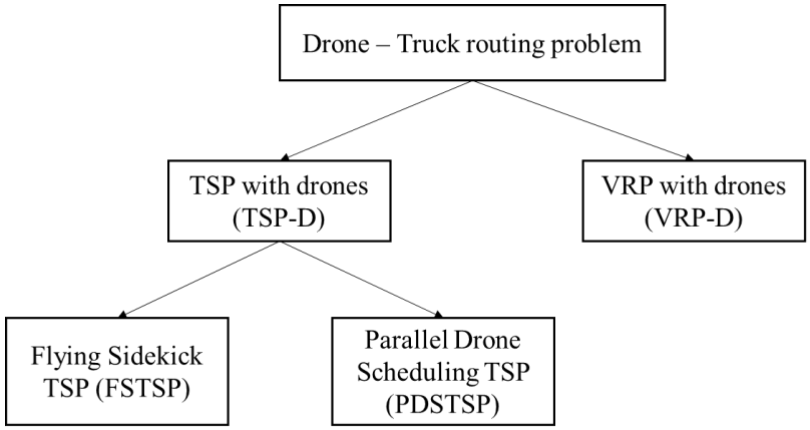


Figure 3.3. Truck - drones routing problems with classification

3.2.1 Flying Sidekick Traveling Salesman Problems

Several models of routing problems have represented the potential situation of drones' implementation within the last-mile delivery system. The earliest model in the context of the "Flying Sidekick Traveling Salesman Problem" (FSTSP) and Parallel Drone Scheduling TSP (PDSTSP) was introduced by Murray and Chu (2015). FSTSP and PDSTSP share the concept of using a truck and a drone(s) to provide services to customers. In the FSTSP drones are launched from the truck while making a delivery and return to the truck in another location, while in the PDSTSP, the truck and drone(s) operate independently, with the truck and the drone(s) delivering to customers in TSP routes from a shared depot. Figure 3.2 illustrate an example of the FSTSP with sortie operations for the drone and Figure 3.3 portrays an example of the PDSTP.

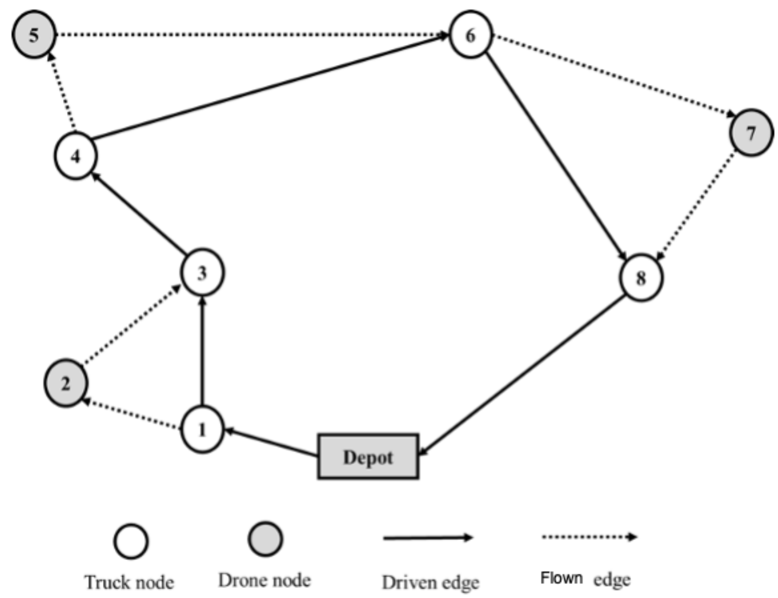


Figure 3.4. An example of FSTSP with drones performing sortie operations with eight customers and one drone.

To reduce the scheduled delivery time, Murray and Chu (2015) proposed a MILP formulation for the FSTSP, in which the drone is assigned to the truck to deliver parcels to clients. In addition, the authors introduced the "Truck First, Drone Second" heuristic approach, in which the truck route is made to address the traveling salesman problem. The truck follows a path that starts at a depot, stops along the way to service customers, and then returns to the depot. The heuristic technique proposed for the PDSTSP, on the other hand, assumes that the drones serve all customers within their maximum range, while the truck serves the remaining customers.

Later, Ponza (2016) proposed a Simulated Annealing (SA) algorithm to solve the FSTSP with up to 200 customers. Additionally, researchers looked at how the drone's durability and peak speed affected the objective values. The findings of numerical experiments showed that increasing these parameters had a favourable effect on the solution's quality, decreasing the tour's duration. Later, Ha et al. (2018), as a study to extend the FSTSP, presented a study in which the drone is able to pick up and deliver items both from trucks and independently from a central depot. Considering multiple trucks and time limit constraints, the authors developed a mathematical formulation minimizing operational costs, including total transportation and waste time costs due to vehicles waiting for the drones. Additionally, to alleviate the computational burden of standard solvers for large instances, two algorithms, namely Iterated Local Search (ILS) and Greedy Randomized Adaptive Search Procedure (GRASP) was proposed.

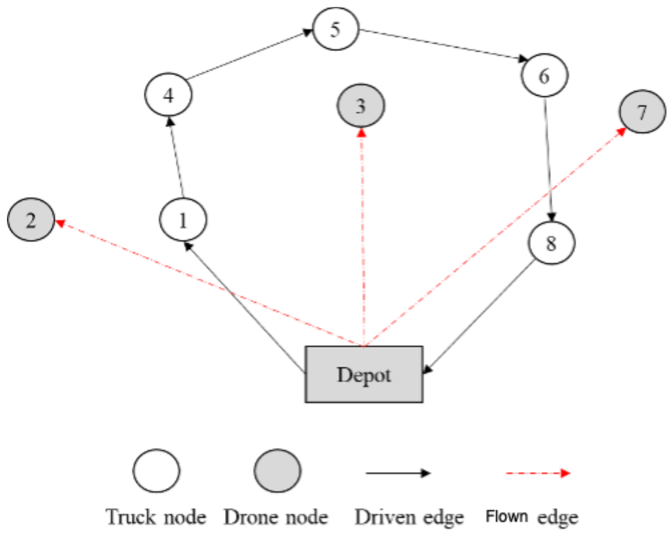


Figure 3.5. An example of PDTSP with eight customers and one drone.

Sacramento et al. (2019) formulated the FSTSP for the capacitated multiple-truck problems with time limit constraints, considering minimizing cost as the objective function. To solve their model for large networks, an Adaptive Large Neighbourhood Search metaheuristic is proposed, the performance of which is analysed with respect to the MIP solutions.

Kitjacharoenchai et al. (2019) investigated the usage of several trucks and numerous unmanned aerial vehicles (UAVs), an extension of the FSTSP, but without endurance limits and launch/delivery time considerations. While UAVs may be launched and collected from multiple vehicles, only one UAV can be launched or retrieved from a customer's site. Additionally, these authors propose a MILP formulation minimizing the arrival time of both drones and trucks at the depot after completing their missions and developed an insertion-based heuristic (ADI) for problems involving up to 50 customers and up to a hundred locations.

Dell’Amico et al. (2021b) updated the FSTSP over the original formulation of Murray and Chu (2015), by improving the objective function to obtain better lower bounds for the problem. Additionally, they considered time windows, capacities, and flying endurance as additional assumptions. Moreover, authors proposed a novel formulation, proven to be better than the original formulation. In another study by Dell'Amico et al. (2021a), an exact algorithm, namely branch and bound (B&B), was designed to efficiently target small-size instances of up to 15 customers. Additionally, the authors proposed a heuristic algorithm, using the branch and bound method to tackle larger instances.

Murray and Raj (2020) extended their study by introducing the multiple flying sidekicks traveling salesman problem (mFSTSP), in which a truck carries a set of heterogeneous drones varying in speed, capacity, service times, and endurance limitations. Endurance is a function of the travel distance without carrying a parcel and the travel distance with a parcel; however, they have also considered different versions with linear and fixed endurance (e.g. a maximum time or a maximum distance). Additional to the previous assumptions, the authors have also considered the scheduling of arrivals and departures of drones, together with a service time at nodes, with drones being able to wait on the ground for the truck to be collected. Nonetheless, depending on the type of drone, only a subset of customers can be served by drones.

Gonzalez-R et al. (2020) proposed a mathematical formulation minimizing the makespan and an Iterated Greedy (IG) algorithm as the solution approach for tackling medium and large size instances for mFSTSP. Later, Windras Mara et al. (2022) presented a mathematical model and developed an Adaptive Large Neighbourhood Search (ALNS) algorithm for mFSTSP. The proposed mathematical formulation also considered the setup time for launching and retrieving the drone. A solver was able to find the optimal solution for instances with up to 8 customers. For larger problems of up to 200 customers, a heuristic approach was able to obtain solutions with acceptable quality.

Murray and Raj (2020) also examine two additional variants of the problem and propose a MILP formulation for each. The first version requires the van driver to manually set up a drone to perform a delivery operation, while the second one assumes automated drone launching and landing. To solve these variants a three-phase iterative heuristic is suggested, in which a TSP is solved heuristically to identify the truck path after splitting the customers into two groups—one to be serviced by the truck and the other by drones. Drone flights are created in the second phase to service the remaining clients. The scheduling is determined in the third step, and local search procedures are used to enhance the solution quality. Based on the results of the numerical experiments, the proposed heuristic solved the problem for up to 100 customers and four drones. In comparison, the MILP could solve the problem to optimality for only up to 8 customers.

Moshref-Javadi et al. (2020) tackle a similar problem by considering a time limit for visiting a customer, where customers are serviced by a truck or a drone. They also have assumed that truck and drone operations are synchronized, with one or more drones mounted on a truck, which serves as a mobile depot. While the vehicle follows a multi-stop route, both drones and the truck may make deliveries; each drone delivers a single product per dispatch. The drones may be released simultaneously from the truck. They proposed a MILP formulation to minimize customer waiting time and compared the results to the conventional truck-only delivery system. The model determines the optimal distribution of customers between the truck and drones, the best truck route, and the optimal launch and retrieving locations of the drones. To solve the problem an ALNS algorithm was proposed, which was able to handle problems involving up to 100 customers and three drones.

(Dell'Amico et al., 2021b) also developed four formulations implemented as branch-and-cut algorithms that improve on earlier algorithms found in the related literature. The authors studied the FSTSP for multiple identical drones with battery restrictions and setup operations when multiple drones are launched or retrieved at the same node. The authors have also examined the impact of having up to five drones available on a single truck with up to 10 customers to be served.

Table 3.1 summarizes the papers reviewed on the FSTSP and the mFSTSP. The table is divided into eight columns; "Article" lists the papers that were reviewed, "Problem" describes the problem type, "Config" characterises the configuration of the fleet ("1" indicates a single and "M" multiple drones/trucks), "Approach" describes the solution method, "Service" indicates whether drone tours cover one or more customers, "Objective" describes the objective function and "Instances" states the maximum number of nodes solved by the proposed solution approach.

Table 3.5. FSTSP and mFSTSP summary table

Article	Problem	Config	Approach	Service	Objective (min)	Instances
Murray and Chu (2015)	FSTSP	1-T; 1- D	PDSTSP heuristic	1- Customer	Completion time	20 nodes

Ponza (2016)	FSTSP	1-T; 1- D	SA	1- Customer	Completion time	Up to 200 nodes
Ha et al. (2018)	FSTSP	1-T; 1- D	ILS; GRASP	1- Customer	Cost	Up to 100 nodes
(Sacramento et al., 2019)	FSTSP	M-T; M-D	ALNS	1- Customer	Cost	Up to 200 nodes
Kitjacharoenchai et al. (2019)	FSTSP	M-T; M-D	ADI	1- Customer	Total time	50 nodes
Gonzalez-R et al. (2020)	mFSTSP	1-T; 1- D	IG	M- Customer	Makespan	Up to 100 nodes
Murray and Raj (2020)	mFSTSP	1-T; M-D	Heuristic	1- Customer	Makespan	Up to 100 nodes
Dell'Amico et al. (2021a)	FSTSP	1-T; 1- D	B & B	1- Customer	Completion time	15 nodes
Dell'Amico et al. (2021b)	FSTSP (with time window)	1-T; 1- D	B & B	1- Customer	Cost	15 nodes

PDSTSP: Parallel Drone Scheduling TSP

3.2.2 Traveling Salesman Problem with Drone (TSP-D)

Agatz et al. (2018) was the first to introduce Traveling Salesman Problem with Drone (TSP-D), which shares its main characteristics with the FSTSP. Similarly, trucks are not being used to transport products in both cases. In fact, the truck simply functions as a mobile depot. In this first platform, the truck must stop at locations known as docking stations where the drone is launched and wait to be received after its mission has been completed. However, the main differences between TSP-D and FSTSP are as follows; the customers can be visited more than once by the truck if it is convenient for launching and collecting a drone, the drone can return to the same node in which it has been launched, drone endurance is unlimited, and launching and rendezvous times are considered as negligible. In other words, TSP-D is distinguished from FSTSP mainly by the ability of the drone to return to its launching location. Agatz et al. (2018) developed a mathematical framework for TSP-D based on a set of feasible operations of truck-drone collaboration. The purpose of the created TSP-D was to reduce the overall completion time of the delivery missions. The authors reported optimum solutions for instances with up to 12 nodes, as well as various rapid heuristics for larger-scale instances.

Bouman et al. (2018) developed exact solution methods based on a dynamic programming approach for TSP-D. In addition, they introduced a 3-pass dynamic programming approach. For the first pass, they adapted the dynamic programming approach for the conventional TSP to find the shortest path for each pair of nodes along the truck path. For the second pass, the authors combined the truck paths with drone nodes to find operations with the least costly way to cover a set of nodes between a given origin-destination (O\D) pair. An optimal TSP-D tour can be constructed if a TSP-D tour will not become longer when an efficient operation is included. The third pass is an extended A* algorithm, which computes the optimal sequence of efficient operations that covers all nodes, having the depot as the start and end point. Furthermore, they restricted the number of locations the truck could visit. According to Bouman et al. (2018), the restriction greatly reduces the overall computing time while not significantly affecting the solution quality. However, the applied restriction could potentially remove the optimal solution from the solution space.

Karak and Abdelghany (2019) presented a MIP model in which a truck, assisted with a drone, delivers purchased items to end customers. In their study, the truck just serves as a mobile depot while the drone reaches the final customer. Additionally, the drone may transport multiple packages during a single delivery operation. Poikonen et al. (2019) proposed four heuristics for TSP-D that were based on the branch and bound algorithm. Throughout this algorithm, each node of the branch-and-bound tree associates with an order to serve a subset of customers. An approximation of the lower bound is proposed at each node by solving a dynamic program.

Dayarian et al. (2020) introduced a novel delivery system concept specially designed to support a highly dynamic environment with tight service guarantees based on using a fleet of drones to resupply delivery trucks. The authors have assumed that resupplying can be done only when a delivery truck is stationary, and a drone can land on the truck's roof. They have also applied a simplification where a single drone and a single truck collaboratively perform deliveries in a small area. Furthermore, they proposed heuristics for each of the two strategies.

Addressing several TSP-D variants, Roberti and Ruthmair (2021) propose a compact mixed-integer linear program (MILP) considering the synchronization constraints between trucks and drones. Furthermore, they implemented dynamic programming recursions in order to model several TSP-D variants and used an exact branch-and-price approach based on a set partitioning formulation. The proposed algorithm was able to solve the problem for up to 39 customers to optimality. Figure 3.4 illustrates a TSP-D example with sortie operation and drone-truck synchronization.

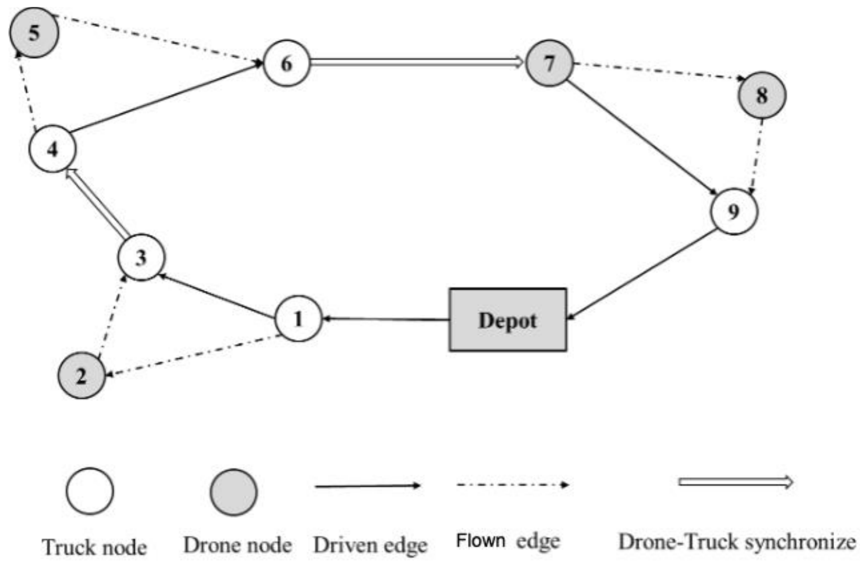


Figure 3.6. A feasible solution of a TSP-D with nine customers and one drone

Traveling salesman problem with multiple drones (TSP-mD) was solved by Tu et al. (2018) by adapting the proposed GRASP heuristic by Ha et al. (2018) and proposing an adaptive large neighbourhood search (ALNS) algorithm. TSP-mD is an extension of the standard TSP with a truck equipped with multiple drones that must serve all the customers to minimize the transportation costs of both the truck and the drones. It is assumed that each node must be visited once, and the drones cannot return to their launching point. Additionally, the battery's endurance is considered, but the drones can wait for the truck on the ground, and thus that time is not included in the battery consumption. The proposed heuristics by Tu et al. (2018) were able to solve the problem for up to 100 nodes and three drones.

Yoon (2018) proposed a MILP model for solving TSP-mD, which includes both fixed and variable costs of deploying multiple drones as well as the truck's operating fuel and labour costs. Additionally, the author performed a sensitivity analysis on several key parameters which affect the costs of truck-drone delivery problems, namely drone speed, drone endurance, number of available drones, truck speed, and geographic customer grid area.

Seifried (2019) proposed a MILP model for the TSP-mD in which all the customers could be served either by the truck or by a set of identical drones. The problem shares the basic assumptions with Murray & Chu (2015), which are as follows; the drones can be launched and retrieved only at the customer locations or at the depot, they serve only one customer at a time, and they cannot return to their launching point. Additionally, it is assumed that drones are at least as fast as the truck, can wait at the nodes without consuming energy, and no scheduling is considered for the launching and rendezvous operations at every node. The author presents a MILP model that solves instances with up to 10 customers and considers up to nine drones.

Table 3.2 summarizes the presented papers in this section for TSP-D and TSP-mD. It should be noted that the table follows the same arrangements of Table 3.1.

Table 3.6. TSP-D and TSP-mD summary table

Article	Problem	Config	Approach	Service	Objective (min)	Instances
Agatz et al. (2018)	TSP-D	1-T; 1- D	Sevral heuristics	1- Customer	Cost	10 nodes
Bouman et al. (2018)	TSP-D	1-T; 1- D	3-pass dynamic programming	1- Customer	Cost	-
Tu et al. (2018)	TSP-mD	1-T; M-D	GRASP; ALNS	1- Customer	Distance	Up to 100 nodes
Yoon (2018)	TSP-mD	1-T; M-D	MILP model	1- Customer	Cost	-
Karak and Abdelghany (2019)	TSP-D	1-T; M-D	VDRH; DDRH	1- Customer	Cost	Up to 100 nodes
Poikonen et al. (2019)	TSP-D	1-T; 1- D	B & B	1- Customer	Makespan	-
Seifried (2019)	TSP-mD	1-T; M-D	MILP model	1- Customer	Makespan	10 nodes
Dayarian et al. (2020)	TSP-D	1-T; 1-D	Heuristic	1- Customer	Cost	-
Roberti and Ruthmair (2021)	TSP-D	1-T; 1-D	Branch & price	1- Customer	Min Makespan	39 nodes
VDRH: vehicle-driven routing heuristic			DDRH: drone-driven routing heuristic			

3.2.3 Vehicle Routing Problem with Drones

The vehicle routing problem with drones (VRP-D) is a generalization of the traveling salesman problem with drone (TSP-D) (Wang et al., 2017), where a set of trucks collaborates with a predetermined number of drones to serve a set of customers (Salama & Srinivas, 2020).

VRP-D involves multiple trucks and drones whose objective varies according to the decision maker's priority. VRP-D was initially proposed by Wang et al. (2017) by considering the following assumptions; each drone can carry only one parcel in each mission, drones have limited endurance, preparation time for each drone launch is negligible, the launching and collecting truck for each drone is the same, the truck can dispatch and pick up a drone only at a node, and the vehicle (truck or drone) that gets to the pickup node first must wait. Wang et al. (2017) aims to minimize the makespan.

Poikonen et al. (2017) extended and strengthened the results of Wang et al. (2017), adding different distance metric for drones and trucks to the original assumptions. (Boysen, Briskorn, et al., 2018) studied the VRP-D where scheduling of deliveries by drones is optimized within a predetermined fixed route of a single truck. In this study, it is assumed that drones may wait for the truck to arrive, in contrast with our study in which trucks must arrive earlier than drones at rendezvous points for the sake of security. Additionally, they have proposed two MILP formulation capable of solving the problem up to 100 customers.

In another study, Schermer et al. (2019) considered a fleet of trucks, where each truck is equipped with a given number of drones, and developed a MILP model minimizing makespan. The authors also proposed a heuristic method in order to cope with the limited performance of the standard solvers when tackling with large instances.

Wang and Sheu (2019) studied the VRP-D and constructed an arc-based model in which the truck does not have to wait at the dispatching location. Rather, trucks can move to a subsequent station and collect the drones. Focusing on the VRP-D with the synchronized truck-drone operation, Kitjacharoenchai et al. (2020) presented a novel routing model that considers a synchronized truck-drone operation by permitting multiple drones to be launched from a truck, servicing one or more customers, and returning to the same truck for a battery change and package collection. The approach handles two levels (echelons) of delivery; primary truck routing from the main depot to service assigned customers, and secondary drone routing from the truck, which acts as an intermediate mobile depot to serve other groups of customers. The model considers the capacity of both trucks and drones to determine optimal routes for trucks and drones that minimize the overall arrival time of both trucks and drones to the depot after delivery completion. Furthermore, the formulated MIP can solve the problem for small-size problems.

Additionally, to tackle large-size problems, authors have developed two metaheuristic algorithms, namely Drone Truck Route Construction (DTRC) and Large Neighbourhood Search (LNS). Leon-Blanco et al. (2022) model presented in Murray and Raj (2020) by allowing drones to serve a set of customers per trip and developed a multi-agent approach to deal with large-size instances.

A more enhanced version of the VRP-D, known as the 'Multi-visit Vehicle-Drone Routing Problem' (MVRP-D), assumes that drones have enough carrying capacity to visit a set of customers per mission. For example, Luo et al. (2021) considered MVRP-D with multi-drones which was formulated as a MILP model minimizing makespan to solve only small problem instances. To address medium and large-size instances, which are more common in real-world circumstances, the authors suggested an enhanced multi-start tabu search (MSTS) algorithm with three major components; a fast approach for evaluating feasibility that can be used at both the drone and solution levels, a construction algorithm based on random sequences to produce initial solutions, and a tabu search algorithm with customized neighbourhood structures. Finally, they were able to solve the problem for up to 100 customers. In another study, Gu et al. (2022) proposed a hybrid iterative local search heuristic with a variable neighbourhood descent procedure (ILS-VND) to solve the MVRP-D with multiple drones, minimizing total cost. The proposed algorithm solved the problem for up to 200 nodes. Figure 3.5 portrays a MTSP-MD example with 15 customers and two drones.

3.2.4 Truck-Drone routing problems with time windows

Few studies have examined the drone-truck routing problem with time windows, despite the fact that taking time windows into account is now necessary for many industries in last mile delivery operations to achieve customer satisfaction. Additionally, considering time windows in logistical problems will add to the complexity of the problem through additional constraints. The drone-truck routing problem with time windows (VRPTWD) was first solved in the research that was carried out by Di Puglia Pugliese & Guerriero (2017) proposing a MILP model. In their scenario, a single truck equipped with a drone delivers the goods to end customers within given time windows. The authors have assumed that the drone can be dispatched multiple times to deliver only one parcel at a time. Ham (2018) added trucks, drones and depots to the parallel drone-truck routing problem, where each client is given a time window. They used drones and trucks to deliver packages separately. Additionally, the authors developed the problem as a constraint programming (CP) model to minimize the maximum completion time throughout all tasks.

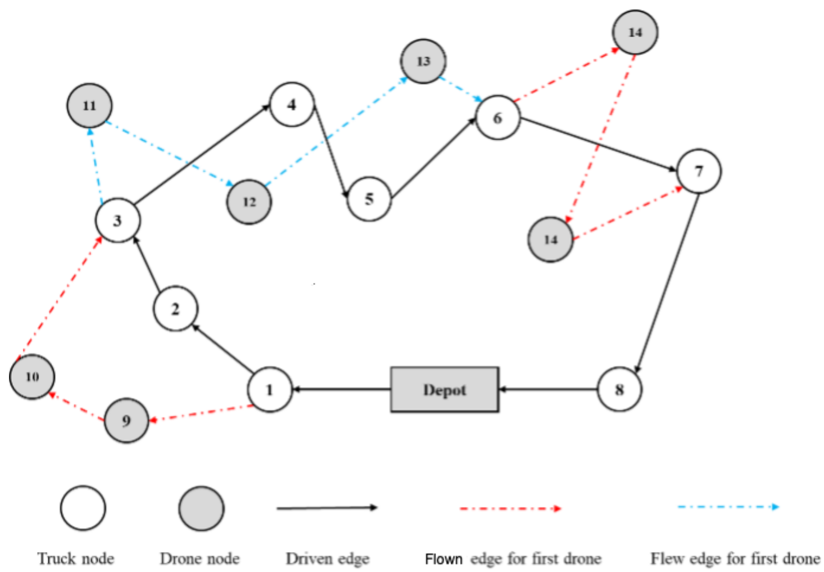


Figure 3.7. An example of MTSP-MD Solution with 15 customers and 2 drones.

Considering a non-linear energy consumption function, Cheng et al. (2020) addressed the drone routing problem (DRP) when each customer is associated with a time window. The authors assumed that a fleet of drones travels to several client locations, and that each drone may stop at many different customers on a single journey. Drones in their research can only be sent out of the depot once. To solve their problem, Cheng et al. (2020) developed a branch-and-cut algorithm to find the optimal routing and scheduling decisions. A recent study from Lan (2020) extended the TSP-D and TSP-mD by considering the time windows constraint (TSPTWD) and developed mathematical formulations for solving up to 7 customers in the given time limit. Additionally, to tackle large-size instances of up to 100 customers the author proposed several heuristics and metaheuristics based on Simulated Annealing (SA) as the solution approach.

In another study Pugliese et al. (2020) provided a qualitative and quantitative analysis of the benefits and drawbacks of drone use in the parcel delivery process by proposing mathematical formulations. Additionally, the authors have also considered time windows and synchronization constraints to take into account realistic scenarios. Di Puglia Pugliese et al.

(2021) studied VRPTWD by proposing a mathematical formulation minimizing cost to tackle small-size problem instances. To handle the complexity of their problem for large instances, a two-phase heuristic embedded in a multi-start framework was developed in which the initial solution was generated by considering only the trucks and a subset of customers. Next, a feasible solution is constructed for the problem at hand by serving the remaining customers with either the trucks or the drones. Each new start of the two-phase heuristic defines a different subset of customers to be served by the truck in phase 1. In Di Puglia Pugliese et al. (2021), it is assumed that time window restrictions for the customers served by the drones are not necessary, meaning that customers associated with time windows are served only by trucks. This assumption, along with the method of solving the scheduling problem, is the main difference of our work.

Coindreau et al. (2021) formulated the VRPTWD as MIP model whose objective seeks to minimize the total costs. They developed an Adaptive Large Neighbourhood Search (ALNS) algorithm to solve the large instances up to 100 customers. They assumed that drone launching, and retrieval times are negligible, which is not realistic. Their results show that truck-and-drone solutions can reduce costs up to 34% compared to traditional truck-only delivery. Kuo et al. (2022) presented a mixed integer programming model addressing the VRPTWD whose objective is to minimize the total cost. To deal with the large instances for which standard solvers are inefficient, they developed a Variable Neighbourhood Search (VNS) algorithm. Two principal considerations differentiate our study from their study: First, in our study, we consider multiple time windows for each customer providing a more general scheduling platform but leading to a substantial increase in problem complexity. Second and more importantly, in the current study, an exact method is developed to schedule customer visits by drones and trucks, unlike Kuo et al. (2022) who applied a metaheuristic method that does not necessarily lead to an optimal solution.

In a recent study Jun Li et al. (2022) introduced the truck and drone routing problem with synchronization by developing a MILP model in which trucks can dispatch or retrieve drones on customer locations or anywhere in between. For simplicity, the authors assumed a fixed velocity for both drones and trucks. Additionally, they have considered a time window for customers to be served, and as the solution approach, an adaptive large neighbourhood search (ALNS) heuristic was proposed to solve the problem of up to 100 nodes.

Table 3.3 contains a summary of the presented papers in this section and follows a similar arrangement to Table 3.1.

Table 3.7. Summary table

Article	Problem	TW	Approach	Service	Objective	Instances
Wang et al. (2017)	M-T; M-D	Single	Worst-case analysis	1- Customer	Completion Time	-
Di Puglia Pugliese and Guerriero (2017)	1-T; 1-D	Single	MSH	1- Customer	Cost	Up to 100 nodes
Ham (2018)	M-T; M-D	Single	Constraint programming	M- Customer	Completion Time	Up to 100 nodes
Wang and Sheu (2019)	M-T; M-D	Single	Exact algorithm	M- Customer	Cost	15 nodes
Cheng et al. (2020)	-T; M-D	Single	B & C	M- Customer	Completion Time	30 nodes
Di Puglia Pugliese et al. (2021)	M-T; M-D	Single	ALNS	1- Customer	Cost	15 nodes
Luo et al. (2021)	1-T; M-D	Single	MSTS	M- Customer	Completion Time	Up to 100 nodes
Coindreau et al. (2021)	M-T; M-D	Single	ALNS	1- Customer	Min cost	Up to 100 nodes
Gu et al. (2022)	M-T; M-D	Single	ILS-VND	M- Customer	Cost	Up to 200 nodes
Kuo et al. (2022)	M-T; M-D	Single	ALNS	1- Customer	Cost	Up to 100 nodes
Li et al. (2022)	M-T; M-D	Single	ALNS	1- Customer	Cost	Up to 100 nodes

MSH: Multi-start Heuristic; TW: Time Window

3.2.5 Drone - Truck synchronization

Effective synchronization between drones and trucks is crucial for the smooth functioning of their operations. A synchronized drone-truck delivery involves having both vehicles present at the same location, coordinating the launch and collection of the drones at the same time. The synchronization plan must account for the necessary time to load/unload drones, replace or charge their batteries, and perform maintenance tasks. Real-world uncertainties or operational constraints may require one vehicle to wait for the other in some synchronized plans. For more information on the synchronization status of drones and trucks, refer to Table 3.4.

Table 3.8.Summary of drone-truck routing problems for synchronization status

Paper	synchronization	Paper	synchronization
Fikar et al. (2016)	No	Ferrandez et al. (2016)	No
Campbell et al. (2017)	Yes	Daknama and Kraus (2017)	Yes
Agatz et al. (2018)	Yes	Carlsson and Song (2018)	Yes
Bouman et al. (2018)	Yes	Chang and Lee (2018)	Yes
Dayarian et al. (2020)	No	Dukkanci et al. (2019)	Yes
Dell'Amico et al. (2021b)	Yes	de Freitas and Penna (2020)	Yes
Gonzalez-R et al. (2020)	Yes	Ha et al. (2018)	Yes
Jeong et al. (2019)	Yes	Karak and Abdelghany (2019)	Yes
Kim and Moon (2018)	Yes	Kitjacharoenchai et al. (2019)	Yes
Li et al. (2020)	No	Luo et al. (2017)	Yes
Marinelli et al. (2018)	Yes	Moshref-Javadi and Lee (2017)	Yes
Murray and Raj (2020)	Yes	Murray and Chu (2015)	Yes/No
bin Othman et al. (2017)	Yes	Peng et al. (2018)	Yes
Poikonen and Golden (2020)	Yes	Poikonen et al. (2019)	Yes
Poikonen et al. (2017)	Yes	Di Puglia Pugliese and Guerriero (2017)	Yes
Sacramento et al. (2019)	Yes	Chung et al. (2020)	Yes
Savuran and Karakaya (2015)	No	Savuran and Karakaya (2016)	No
Schermer et al. (2018)	Yes	Schermer et al. (2019a)	Yes
Schermer et al. (2019b)	Yes	Tu et al. (2018)	Yes
Ulmer and Thomas (2018)	No	Wang et al. (2017)	Yes
Wang and Sheu (2019)	No	Yoon (2018)	Yes
Yurek and Ozmutlu (2018)	Yes	Ham (2018)	No
Mbiadou Saleu et al. (2018)	No	Dell'Amico et al. (2020)	No

This chapter proposes an *m*-drone, *n*-truck routing framework which allows the drones to be retrieved by any nearby truck which is not necessarily its mother truck, leading to a flexible cooperation between drones and trucks. We also allow various drone dispatch and collection strategies including dispatch-wait-collect and dispatch-move-collect.

Chapter 4 introduces another truck-drone routing platform that incorporates synchronization constraints and time windows.

3.3 PROBLEM DEFINITION

The problem under investigation in this chapter is the delivery of parcels, where multiple trucks and multiple drones co-deliver a set of packages from a local depot to a set of customers. The proposed model is inherently a vehicle routing problem (VRP) because vehicle capacities are considered, the solution of which requires the construction of optimal interconnected truck and drone routes.

The aim is to minimize the objective function which is the total logistic cost. The trucks start their tours from a depot with a number of drones mounted on their roofs. Trucks may travel directly to end customer locations to deliver packages whilst also acting as mobile depots for a fleet of drones. This cooperation will allow more customers to be served and eliminate the need to establish fixed warehouses in places close to customer locations. By way of example, Figure 3.6 shows a network consisting of two trucks assisted by drones delivering goods to customers.

Depending on the size and weight of the packages, drones can visit one or more customers on each tour. The drones are loaded and dispatched to demand points, but the drones do not have to return to the location of dispatching or return to the same truck after a delivery mission.

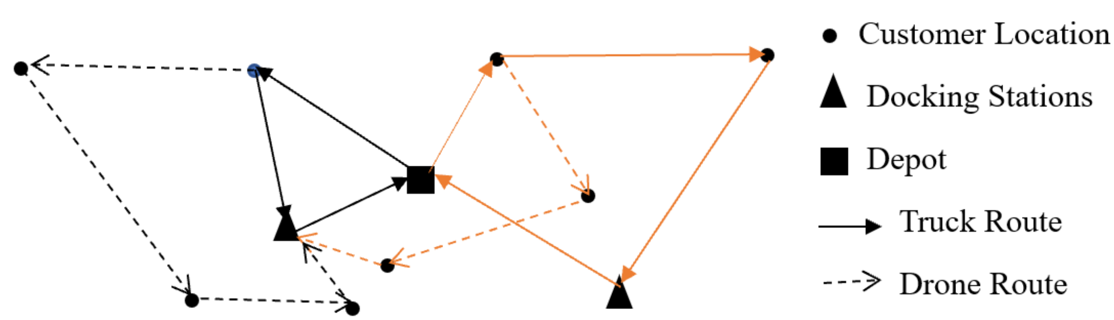


Figure 3.6. An example of a flexible truck-drone network

Instead, trucks can continue their routes and deliver packages and recover drones that were not necessarily dispatched by themselves. At designated points known as docking stations, trucks are responsible for retrieving drones, creating a stop where both the drones and trucks can pause and wait for each other.

3.3.1 Assumptions

The following assumptions are considered for modelling this problem:

- Limited Truck Capacity for Drones: To ensure operational efficiency, trucks are constrained by a limited drone-carrying capacity, mirroring real-world constraints on vehicle capabilities.
- Dual Delivery Capability: Allowing both trucks and drones to deliver orders reflects the need for a versatile delivery system that can adapt to different package specifications and customer locations.
- Package Assignment Based on Specifications: Assigning delivery tasks to trucks or drones based on package specifications and customer location ensures efficient and practical order fulfillment, optimizing resource use.
- Drone Pickup at Docking Stations: Confining drone pickup to docking stations aligns with the operational realities of drone charging and package loading.
- Flexible Drone Landings: Allowing drones to land on any vehicle, even those from which they were not dispatched, maximizes operational flexibility, enhancing delivery efficiency.
- Versatile Drone Dispatch Points: Drones can be dispatched from both customer locations and docking stations, expanding the range of deployment options and improving responsiveness.
- Return to Depot for Trucks: Requiring trucks to return to the depot after completing their mission ensures orderly fleet management and resource allocation.
- Dependent Drone Returns: Drones rely on trucks to return to the depot, reflecting the need for a centralized retrieval system and simplifying logistics planning.

- Exclusivity of Truck or Drone Delivery: Assigning either a truck or drone for customer delivery prevents redundant delivery attempts, streamlining the delivery process and ensuring order accuracy.

3.3.2 Parameters and variables

The notations used to formulate the mathematical model are as follows:

Sets

N	Set of all nodes, indexed by $i, j, l, m, n, k \in \{0,1, \dots, N\}$;
N_D:	Subset of nodes including the depot, represented by 0;
N_v:	Subset of nodes representing the nodes at which trucks can stop;
N_c:	Subset of nodes including the customer locations;
N_s:	Subset of nodes representing the docking stations at which drones can be collected;
V:	Set of trucks, indexed by $v \in V$;
D:	Set of drones, indexed by $d \in D$;

Parameters

max_v:	Maximum number of drones that vehicle $v \in V$ can carry;
cv_{ij}:	The travel cost of vehicles moving from node $i \in N$ to node $j \in N$;
cd_{ml}:	The travel cost of drones flying from node $m \in N_V$ to node $l \in N_V$;
cwv:	Capacity of vehicle;
cwd:	Load lifting capacity of drones;
w_m:	Delivery weight of customer $m \in N_C$;
$dist_{ml}$:	The length of link $(m, l) \in N$;
r_d:	Flight range of drone $d \in D$;
$maxd$:	Maximum number of drones carried by each truck;

Variables

das_{dimj}:	Binary variable which takes 1 if drone $d \in D$ contains the demand of customer $m \in N_C$ and travels link $(i, m) \in N_V$ when dispatched from node $i \in N_V$ and joins a truck at node $j \in N_s$;

das^1_{dimj}:	Binary variable which takes 1 if node $m \in N_C$ represents the only customer location visited by a drone in a tour which is dispatched from nod $i \in N_V$ and collected at nod , $j \in N_S$ and otherwise 0;
vas_{vm}:	Binary variable which takes 1 if vehicle$v \in V$ contains the demand of customer $m \in N_C$;
dw_{dij}:	Pick-up load carried by drone $d \in D$ dispatching from node $i \in N_S$ and collected in node$j \in N_S$;
vw_{vi}:	Pick-up load carried by vehicle $v \in V$ moving from depot toward node $i \in N_V$;
dst_{dij}:	The distance travelled by drone $d \in D$ dispatched from node $i \in N_V$and collected in node$j \in N_S$;
b_{dij}:	Binary variable which takes 1 if drone$d \in D$ dispatched from node $i \in N_v$ is collected at station $j \in N_s$, and 0 otherwise;
f_{vj}:	Binary variable which takes 1 if drone $d \in D$is dispatched from vehicle $v \in V$at station $j \in N_V$, and 0 otherwise;
u_{vi}:	Specifies the order of node $i \in N_V$ in the route of vehicle $v \in V$;
y_{vij}:	Binary variable which takes 1 if truck $v \in V$at moves from node $i \in N_V$ to node $j \in N_V$, and 0 otherwise;
z_{dilmj}:	Binary variable which takes 1 if drone$d \in D$ dispatched from node $i \in N_v$ and collected at station $j \in N_s$ flies from node $l \in N_v$ to $m \in N_v$, and 0 otherwise;

The problem under investigation addresses hybrid truck-drone delivery route planning, where $G(N, A)$ represents the network consisting of N_V nodes and A_V links. The set of nodes comprises a single depot $N_D = \{0\}$, docking stations N_S, and customer points N_C. The nodes to which trucks can travel to are $N_V = N_D \cup N_s \cup N_c$. A set of drones represented by D, in tandem with a set of trucks denoted by V perform the delivery tasks. The trucks are able to carry a small number of drones at the start of their tour from a depot. Drones can fly toward customer points N_C, either independently or in tandem with trucks. Each customer's delivery task is assigned to a truck or a drone, depending on demand and location. If a customer is assigned to a truck, a drone must not fly to that customer and vice versa. After completing their delivery tasks, drones can land at docking stations to be picked up by a nearby truck. Note that drones can be sent to customers both from other customer locations and from docking stations. However, drones dispatched at customer locations, after finishing their mission, must land at docking stations and wait for a truck. Like trucks, drones are assumed to be capable of visiting multiple customers per mission. The aim of the solution process is to find the optimum routing and assignment decisions such that the total logistics cost is minimized.

3.4 MATHEMATICAL FORMULATION

In this section, a MIP model is developed to describe the cooperative system of trucks and drones.

$$\sum_{v\in V}\sum_{i\in N_V}\sum_{j\in N_V} y_{vij}\, cv_{ij} + \sum_{i\in N_V}\sum_{m\in N_C}\sum_{l\in N_V}\sum_{j\in N_S} z_{dilmj} cd_{ml} + TC\sum_{v\in V} g_v + SC\sum_{j\in N_S} s_j \tag{1}$$

$$\sum_{v\in V} y_{v0j} \leq 1 \qquad \forall j \in N_V, \tag{2}$$

$$y_{vij} \leq \sum_{k\in N_V} y_{v0k} \qquad \forall i, j, \in N_V: i \neq j, v \in V \tag{3}$$

$$\sum_{i\in N_V: i\neq j}\sum_{v\in V} y_{vij} = \sum_{i\in N_V: i\neq j}\sum_{v\in V} y_{vji} \qquad \forall j \in N_V, \tag{4}$$

$$z_{dimlj} + z_{dilmj} \leq 1 \qquad \forall d \in D, i \in N_V, j \in N_S, m \in N_V, l \in N_V: l \neq i, l \neq m, l \neq j, m \neq i, \tag{5}$$

$$\sum_{j\in N_S, i\neq j}\sum_{i\in N_V, i\neq j}\sum_{d\in D} z_{dilmj} + \sum_{j\in N_S, i\neq j}\sum_{i\in N_V, i\neq j}\sum_{d\in D} z_{dimlj} + \sum_{v\in V} y_{vml} \leq 1 \qquad \forall l, m \in N_V. m \neq l \tag{6}$$

$$\sum_{m\in N_V, m\neq l} z_{dilmj} = 0 \qquad \forall i \in N_V, l \in N_S, j \in N_S: l \neq j, \qquad (7)$$

$$\sum_{r\in N_C, r\neq m, r\neq j} \sum_{k\in N_C, k\neq r, k\neq i, k\neq m} z_{dikrj} \leq \vee\, N_v \vee \left(2 - z_{diimj} - z_{dimjj} + das_{dimj}\right) \qquad \forall d \in D, i \in N_V, j \in N_S, m \in N_C: m \neq i, m \neq j \qquad (8)$$

$$\sum_{m\in N_C} \sum_{l\in N_C: l\neq m} z_{dilmj} \leq \sum_{m\in N_C: m\neq i} das_{dimj} \qquad \forall d \in D, i \in N_V, j \in N_S, \qquad (9)$$

$$\sum_{m\in N_V: m\neq l, m\neq j, l\neq i, l\neq j} z_{dimlj} = \sum_{m\in N_V: m\neq l, m\neq i, l\neq j, l\neq i} z_{dilmj} \qquad \forall i \in N_V, l \in N_V, j \in N_S, d \in D, i \neq j \qquad (10)$$

$$\sum_{m\in N_V: m\neq l, l\neq i} z_{dimli} = \sum_{m\in N_V: m\neq l, l\neq i} z_{dilmi} \qquad \forall i \in N_V, l \in N_V, j \in N_S, d \in D, i = j \qquad (11)$$

$$\sum_{m\in N_V} \sum_{l\in N_V: l\neq m} z_{dimlj} \leq |N_V| \sum_{m\in N_V: m\neq j} z_{dimjj} \qquad \forall (i,j) \in N_V, d \in D, \qquad (12)$$

$$\sum_{m\in N_V} \sum_{l\in N_V: m\neq l} z_{dimlj} \leq |N_V| \sum_{l\in N_V: l\neq i} z_{diilj} \qquad \forall i, j \in Nv, d \in D, \qquad (13)$$

$$\sum_{l\in N_V, l\neq i} z_{diilj} \leq |N_V| \sum_{k\in N: k\neq i} \sum_{v\in V} y_{vki} \qquad \forall i \in N_V, j \in N_S, d \in D, \qquad (14)$$

$$\sum_{m\in N_V, m\neq j} z_{dimjj} \leq |N_V| \sum_{k\in N: k\neq j} \sum_{v\in V} y_{vkj} \qquad \forall i \in N_V, j \in N_S, d \in D, \qquad (15)$$

$$\sum_{d\in D} \sum_{i\in N_V} \sum_{j\in N_S} \left(das_{dimj} + das^1_{dimj}\right) + \sum_{v} vas_{vm} = 1 \qquad \forall m \in N_C, \qquad (16)$$

$$\sum_{l\in N: m\neq l} y_{vlm} = vas_{vm} \qquad \forall m \in N_C, v \in V, \qquad (17)$$

$$\sum_{l, l\neq m} z_{dilmj} = das_{dimj} + das^1_{dimj} \qquad \forall i \in N_V, j \in N_S, m \in N_C, d \in D \qquad (18)$$

$$u_{vi} - u_{vj} + |N| y_{vij} \leq |N| - 1 \qquad \forall i \in N, j \in N, v \in V \qquad (19)$$

$$u_{v0} = 1 \qquad \forall v \in V \tag{20}$$

$$dw_{dimj} \leq cwd \qquad \forall (i,m) \in N_V, j \in N_S, d \in D \tag{21}$$

$$vw_{vi} + \sum_{m \in N_C} vas_{vm}\, w_m \leq cwv \qquad \forall i \in N, v \in V \tag{22}$$

$$dw_{dilj} \geq dw_{dimj} + w_m - M\left(1 - z_{dimlj}\right) \qquad \forall (i,l,m) \in N_V, j \in N_S, d \in D \tag{23}$$

$$dw_{dilj} \leq dw_{dimj} + w_m + M\left(1 - z_{dimlj}\right) \qquad \forall (i,l,m) \in N_V, j \in N_S, d \in D \tag{24}$$

$$vw_{vj} \geq vw_{vi} + \sum_{k \in N_V} dw_{dik} - M\left(1 - y_{vij}\right) \qquad \forall i \in N_V, j \in N, v \in V \tag{25}$$

$$vw_{vj} \leq vw_{vi} + \sum_{k \in N_V} dw_{dik} + M\left(1 - y_{vij}\right) \qquad \forall i \in N_V, j \in N, v \in V \tag{26}$$

$$dst_{dilj} \geq dst_{dimj} + dist_{lm} - M\left(1 - z_{dimlj}\right) \qquad \forall (i,m,l) \in N_V, j \in N_S, d \in D \tag{27}$$

$$dst_{dilj} \leq dst_{dimj} + dist_{lm} + M\left(1 - z_{dimlj}\right) \qquad \forall (i,m,l) \in N_V, j \in N_S, d \in D \tag{28}$$

$$dst_{dilj} \leq r_d \qquad \forall (i,m) \in N_V, j \in N_S, d \in D \tag{29}$$

$$2|N_V| b_{dij} \geq \sum_{m \in N_C : m \neq i} das_{dimj} \qquad \forall i \in N_V, j \in N_S, d \in D \tag{30}$$

$$b_{dij} \leq \sum_{m \in N_C : m \neq i} das_{dimj} \qquad \forall i \in N_V, j \in N_S, d \in D \tag{31}$$

$$f_{vj} \leq \sum_{d} \sum_{k \in N_V} b_{dkj} - \sum_{d} \sum_{k \in N_V} b_{djk} + f_{vi} + |D|\left(1 - y_{vij}\right) \qquad \forall i \in N, j \in N\{0\}, v \in V : i \neq j \tag{32}$$

$$f_{vj} \geq \sum_{d} \sum_{k \in Nv} b_{dkj} - \sum_{d} \sum_{k \in Nv} b_{djk} + f_{vi} - |D|\left(1 - y_{vij}\right) \qquad \forall i \in N, j \in N\{0\}, v \in V : i \neq j \tag{33}$$

$$\sum_{j \in N} vw_{vj} \leq cwv \sum_{i \in N, i \neq j} \sum_{j \in N} y_{vij} \qquad \forall v \in V \tag{34}$$

$$\sum_{i \in N_V} y_{v0i} \leq |N_S| g_v \qquad \forall v \in V \tag{35}$$

$$z_{diimj} + z_{dimjj} - 1 \leq das_{dimj} \quad \forall i \in N_V, m \in N_C, j \in N_S: i \neq m \quad (36)$$

$$z_{diimj} + z_{dimjj} \geq 2\left(das_{dimj}\right) \quad \forall i \in N_V, m \in N_C, j \in N_S: i \neq m \quad (37)$$

$$\sum_{i \in N_V: i \neq j} \sum_{v \in V} y_{vij} \leq |N_V| S_j \quad \forall j \in N_S \quad (38)$$

$$dw_{dij}, vw_{vi}, dst_{dij}, u_{vi} \geq 0; y_{vij}, z_{dilmj}, das_{dimj}, vas_{vm}, b_{dij}, f_{vj} \in \{0,1\} \quad (38)$$

In the proposed model, the objective function is the sum of operating costs of the drones and trucks; the cost of maintaining the depot and docking stations; and the fixed cost of purchasing the drones and trucks, which is considered as a fixed rental cost per day to be dimensionally consistent with other operating costs in the objective function Equation (2) requires that each truck leave the depot only once. Constraint (3) ensures that a truck can build a route only if it leaves the depot. Equation (4) ensures route continuity for the truck. Constraint (5) removes sub-tours for drones. Constraint (6) requires that the route between two customer nodes be traversed either by drones or by trucks. Constraint (7) ensures that drones do not fly to stations that are neither its origin nor its destination. Constraint (8) requires the drone to either form a route or visit a customer as a back-and-forth trip. Constraint (9) specifies which customers are met by the drones. Constraints (10) and (11) guarantee route continuity for a drone. Constraints (12) and (13) ensure that a drone can create a route only when it is dispatched from a point and collected at a docking station. Constraints (14) and (15) guarantee that a drone can fly from a node if a truck is present at that node. Constraint (16) ensures that all customers are visited either by drones or by trucks. Constraint (17) specifies the customers who are visited by trucks are not visited by drones. Constraint (18) determines whether a drone visits a customer on a back-and-forth trip. Constraints (19) and (20) are the sub-tour elimination equations for trucks. Constraints (21) and (22) specify the maximum capacity of drones and trucks, respectively. Constraints (23) and (24) specifies the amount of cargo that may be carried by a drone between two nodes. Likewise, constraints (25) and (26) determine the amount of cargo that may be carried by a truck between two nodes. Constraints (27) and (28) set the distance travelled by a drone. Constraint (29) ensures that the distance travelled by a drone does not exceed its flight range. Constraints (30) and (31) specify which nodes a drone flies from, and at which docking stations it is retrieved. Constraints (32) and (33) quantify the number of drones flown from and collected by each truck. Constraint (34) ensures that the weight of cargo assigned to drones on a truck does not exceed the carrying capacity of a truck. Constraint (35) indicates the number of trucks used in the network. Constraints (36) and (37) specify drones that visit customers by back-and-forth trips. Constraint (38) determines the number of docking stations established in the network. Constraint (39) specifies the domain of the variables used in the formulation.

3.5 SOLUTION APPROACH

3.5.1 Constructing the initial solution: Clarke & Wright savings heuristic

This chapter presents two heuristic algorithms for solving large-scale problem instances. The need to provide an efficient solution method to solve the problem in a reasonable execution time is discussed experimentally in Section 3.6. All algorithms presented in this study use network geometric information and customer information as input. In this chapter we use the Clarke & Wright savings heuristic (CWH) to generate routes (Clarke & Wright, 1964). Our proposed algorithms remove or add candidate points to be visited by the truck during the solution procedure. If a candidate node identified for removal is a customer node, this node is entered into the set of points to be serviced by drones. Otherwise, if the candidate node identified for removal is a docking station, then it is ignored by both trucks and drones. To improve the performance of the proposed algorithms, novel heuristic methods are designed to produce high-quality feasible solutions. These algorithms improve on the routes generated by CWH and produce shorter routes. To better understand the steps applied, the algorithm’s procedures are illustrated for a small-size instance. In all the figures depicted in this study, circles represent customers, triangles represent stations, and squares represents depots.

The choice of the Clarke & Wright savings heuristic (CWH) as the basis of our solution method can be justified for several reasons:

- Simplicity and Efficiency: The CWH is known for its simplicity and computational efficiency. It does not require complex mathematical formulations or extensive computational resources, making it well-suited for solving large-scale problem instances within a reasonable execution time.
- Versatility: CWH is versatile and can be adapted to various routing scenarios, including those involving multiple vehicles with different capabilities, such as trucks and drones. Its flexibility in handling different types of nodes (customers, stations, depots) and incorporating them into the routing solutions makes it suitable for this problem's complexity.
- Effective Initial Solutions: CWH's ability to generate initial routing solutions efficiently is crucial in the context of our problem. These initial solutions can serve as a solid foundation for further optimization and fine-tuning.
- Resource Allocation: CWH can effectively allocate customers and stations to trucks and drones based on their specifications and locations. This reflects the need for an intelligent and practical delivery system that optimizes resource utilization while ensuring efficient order fulfillment.

In this section, we first provide an overview of CWH. The logic of CWH is that it first generates small back-and-forth routes as primary routes. Figure 3.7 shows a simple example of these back-and-forth routes, consisting of a depot and four stations.

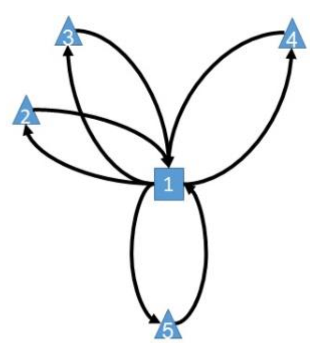


Figure 3.7. Back-and-forth routes generated by CWH

It then calculates the savings resulted from merging each pair of back-and-forth routes and merges the pairs with the highest obtained saving. This continues until routes can no longer be merged due to infeasibility or incurred cost. The pair saving for customers is calculated in CWH according to equation (40). Assume that nodes (2) and (3) shown in the example have the least pair saving, then the new routes will be as shown on the right side of Figure 3.8.

To calculate the cost of merging a pair of customer nodes, we first calculate the values of each pair saving (ϑ_{ij}) using equations. (40) and (41) and store them in ψ_D.

$$l_{is_i} = min\left(l_{is_k} \forall s_k \in N_V\right) \tag{40}$$

$$\vartheta_{ij} = l_{is_i} + l_{js_j} - l_{ij} \tag{41}$$

In general, the algorithms presented in this study remove the docking stations one by one from N_S and then identify the nodes to be visited by trucks. As the nodes assigned to the truck are identified, the nodes that the drones must visit are also determined.

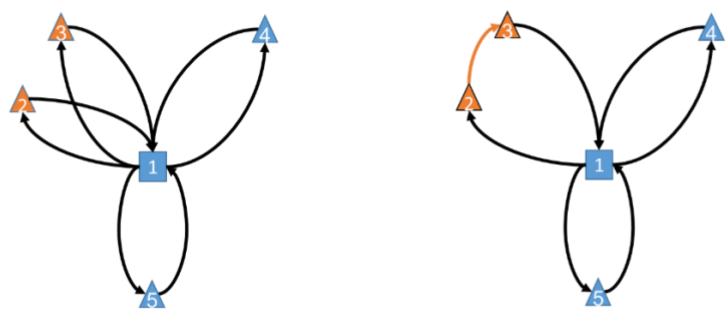


Figure 3.8. Merging pairs with the greatest saving

The proposed algorithms consist of an inner loop and an outer loop. The outer loop determines the stations in N_S, while the inner loop identifies the nodes to be included in N_V. Figure 3.9 depicts the flowchart of the outer loop.

During the solution procedure, the nodes in N_S are eliminated one by one, and the total cost after each elimination is calculated. In the algorithms, the change in the objective function induced by each elimination is evaluated relative to the previous state.

Since at first all stations exist, it is necessary to evaluate whether a removal would result in a better solution. Thus, a null member, representing the initial situation, is added to N_S. The station whose removal produces the least total cost is eliminated from N_S. This process will continue until either at least one station remains on the network, or the removal of stations does not result in a lower cost. H0 presents the steps in the main body of our two algorithms. Note that the N_S must include at least one station (N_S must be greater than 1) since at least one station is needed at which drones can be retrieved by trucks.

H0: The Clarke and Wright Savings Heuristic

Input: Network topology and customers' information
Result: $FinalRoutes, BestCost$
$continue \leftarrow$ true
$FinalRoutes \leftarrow \emptyset$
$BestCost \leftarrow +\infty$
$BestRemove \leftarrow \emptyset$
$CurrentCost \leftarrow +\infty$
$N_S' \leftarrow N_S$
While ($continue$= true) do
    $continue \leftarrow$ false
    If (the number of stations in N_S>1) then
        $N_S' \leftarrow N_S$
        For each ($i \in N_S$) do
            Remove i from N_S'
            $[NewCost, NewRoutes] \leftarrow$BFCWH($N_S'$)
            If ($NewCost < BestCost$) then
                $BestCost \leftarrow NewCost$
                $FinalRoutes \leftarrow NewRoutes$
                $BestRemove \leftarrow i$
                $continue \leftarrow$ true
            End if
            Add i to N_S'
        End for
    End if
    Remove $BestRemove$ from N_S
End while
Return $FinalRoutes, BestCost$

In the inner loop, routes are constructed according to the N_S and N_V and are further improved by our proposed heuristics. Figure 3.10 shows the flowchart of the inner loop.

Constructing the drone routes

First, according to the logic of CWH, drone routes are formed. For this purpose, considering N_V nodes, pair saving for available customers for drones is calculated using equations (40) and (41) and is saved as ψ_D. An initial route is then formed for each customer point available to the drones. These routes start from the nearest point to be visited by trucks, ending up at the nearest station after the designated customer points are visited by drones. The sub-figures a-d, presented in Figure 3.11, show the steps of constructing the initial routes for a network with five customers and three stations.

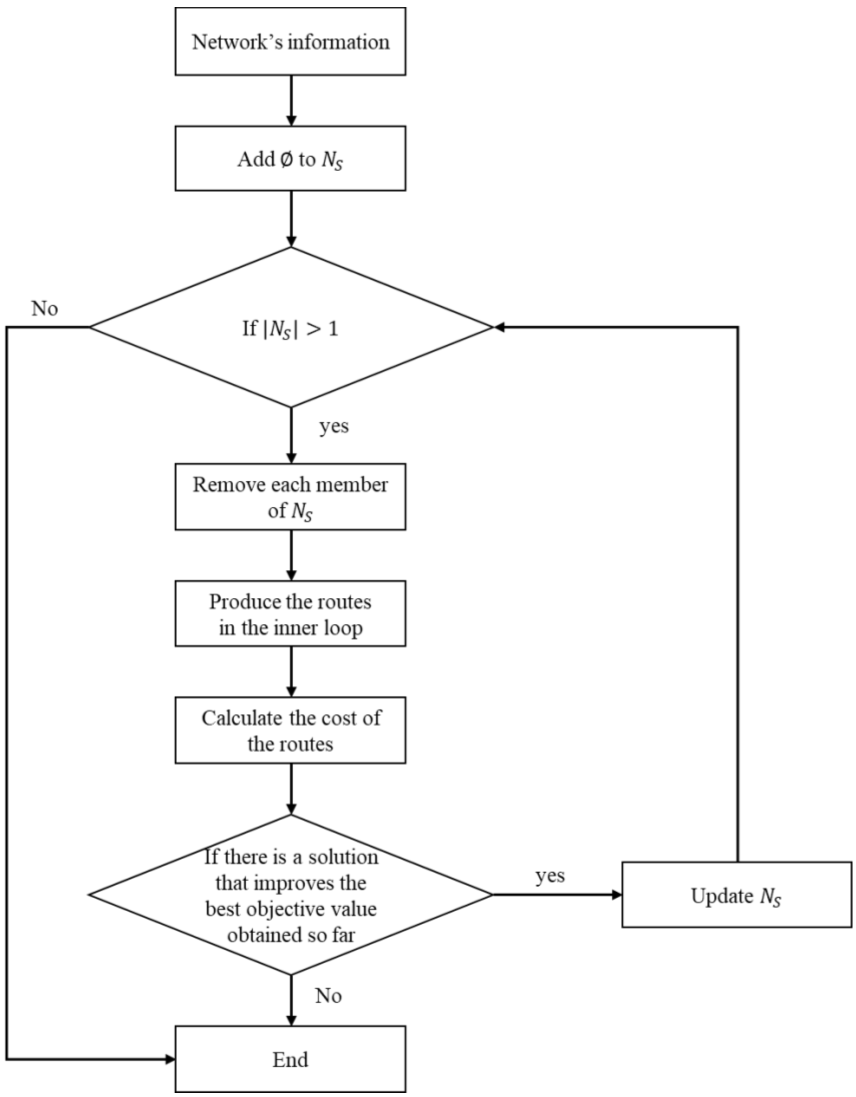


Figure 3.9. Flowchart of the outer loop in the proposed heuristic

Calculating ψ_D associated with routes generated, we merge the routes following H1.

H1: Build_Drone_Routes

Input: ψ_D, N_V

Result: S_D

$S_D =$ Calculate initial drone routes using N_V

Sort ψ_D descending

while $\psi_D \neq \emptyset$ do

m, n ←pick the two nodes from the first element of ψ_D which the corresponding ϑ_{mn} has the highest value

Get $route_1$ that contains customer m, and $route_2$ that contains customer n from S_D
if ($route_1 \neq route_2$ & customers m and n are not intermediate nodes) then
  if (m is the first edge in $route_1$ and n is the last edge in $route_2$) then
    $merged \leftarrow$ Merge customers m and n in a new route, which originates from the origin of $route_1$ and ends at the destination of $route_2$.
  end if
  if $merged \neq \emptyset$ then
    $ImprovedRoute$ ←DistanceImprove ($merged$)
    Remove $route_1$ and $route_2$ from S_D and add $ImprovedRoute$ to S_D
    if (the $ImprovedRoute$is NOT feasible) then
      Remove $ImprovedRoute$ from S_D, and add $route_1$ and $route_2$ to S_D
    end if
  end if
end if
Eliminate ϑ_{mn} from ψ_D
end while
Return S_D

For instance, the routes merged in the network illustrated in Figure 3.11(d) is depicted in Figure 3.12.

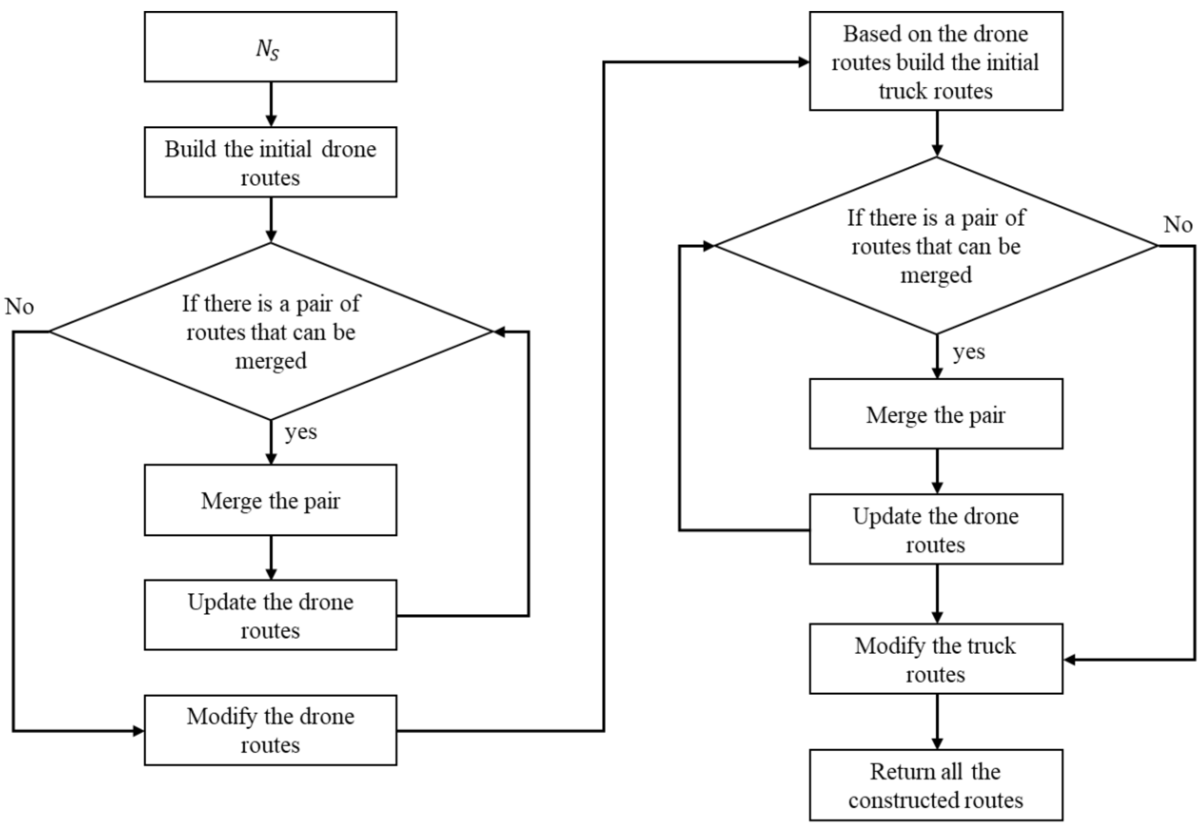


Figure 3.10. Flowchart of the inner loop

Modifying the drone routes

After the drone routes are created, they are modified according to the H2 algorithm. The algorithm is initialized by identifying all the points where trucks have dispatched or recovered more than $maxd$ drones. Then we change the origin of the routes where the number of drones dispatched or received exceeds the carrying capacity of the truck to the nearest available point (except for the current origin) and save the distance travelled to the new origin. The stations for which the number of incoming drones exceeds the carrying capacity of trucks are thus identified. Then, the destinations of the routes entering this station are changed to the nearest station with sufficient spare capacity. If there is not a station with sufficient capacity, we set the origin of the route to the current destination station. Then we calculate and save the new route length obtained by making this change. After performing all these steps for all nodes for which the drone carrying capacity of trucks drones is violated, we select the nearest station. This process is continued until all trucks have met their drone load limit ($maxd$).

In the network depicted in Figure 3.12(a), the capacity of node 4 is violated with the assumption of $maxd = 2$. Figures 3.12(b), (c), and (d) show the steps, described in H2, required to generate a feasible route.

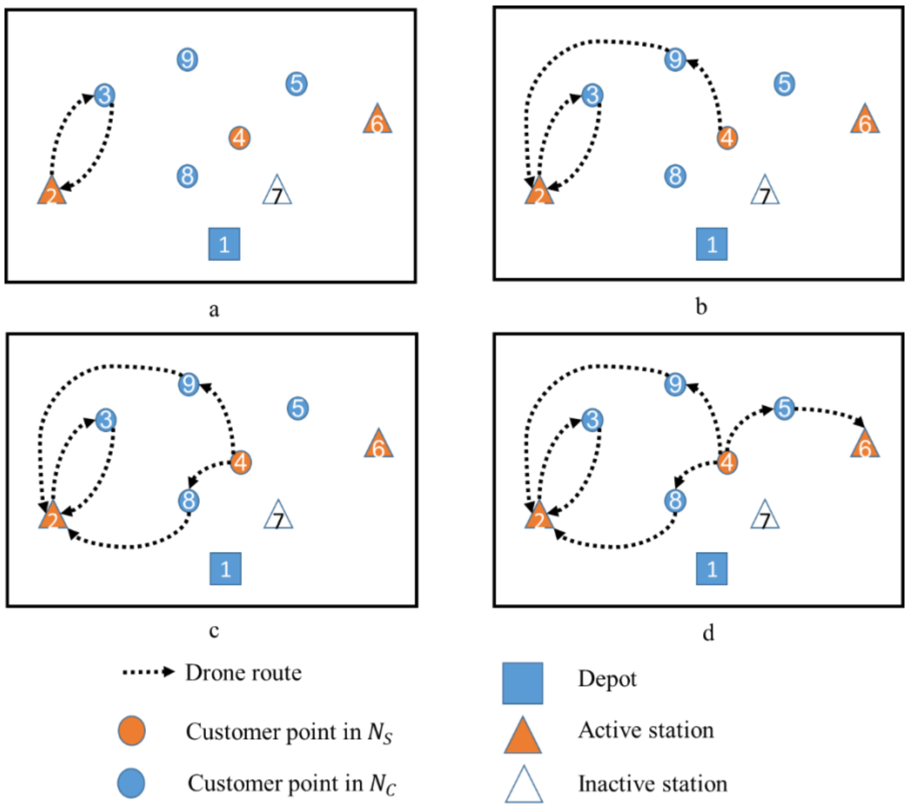


Figure 3.11. Initial routes for a network with five customers and three stations.

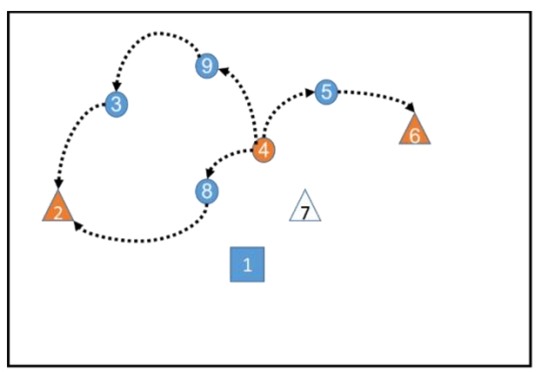


Figure 3.12. The routes merged in the network illustrated in Figure 6(d)

H2: Imrove_Drone_Routes

Input: S_D, N_V, N_S
Result: S_D
$CollectionViolated$ ← the nodes where the number of collected drones exceeds $maxd$.
$DispatchViolated$ ← the nodes where the number of dispatched drones exceeds $maxd$.
$continue$ ← true
$saving$ ←0
$move$ ← ∅

While ($DispatchViolated$ and $CollectionViolated \neq \emptyset$) do
For each ($i \in CollectionViolated$) do
For each (drone route $sd \in SD$ which ends at i) do
If (i is not the origin of sd) then
If (there is a node in $N_V/\{i\}$ which has capacity to collect another drone) then
$sd' \leftarrow$ change the destination of sd to a node in $N_V/\{i\}$ which is closest to the last visited costumer in sd and which has capacity to collect another drone.
Else
$sd' \leftarrow$ change the destination of sd to its origin node and make it a close loop.
End if
$saving \leftarrow$ length of $sd -$ length of sd'
Add (sd,sd',$saving$) to $move$
End for
End for
For each ($i \in DispatchViolated$) do
For each (drone route sd which starts form i) do
If (i is not the destination of sd) then
If (there is a node in $N_V/\{i\}$ which has capacity to dispatch another drone) then
$sd' \leftarrow$ change the origin of sd to a node in $N_V/\{i\}$ which is closest to the first visited costumer in sd and which has capacity to dispatch another drone.
Else
$sd' \leftarrow$ change the origin of sd to its destination node and make it a close loop.
End if
$saving \leftarrow$ length of $sd -$ length of sd'
Add (sd,sd',$saving$) to $move$
End for
End for
According to $move$, apply the change that results the highest saving.
Update SD, $DispatchViolated$, and$CollectionViolated$ based on the change.
End while
Return the SD

From these three states, the routes shown in Figure 3.13(d) are selected as the best because they are the shortest. The drone capacity for the other nodes in Figure 3.13(d) is met so the routes shown in Figure 3.13(d) will be output by the H2 function.

Constructing truck routes based on drone routes

After modifying the drone routes, the calculations for the nodes are updated. According to the same process carried out for drone routes, we calculate the pair saving associated with nodes to be visited by trucks and save them as ψ_V. The initial truck routes are then created, which includes round-trip routes from the depot to the nodes existing in N_V. These routes are constructed based on ψ_V following the H3 function.

H3: Build__Routes

Input: S_D,ψ_V
Result: S_V
$S_V \leftarrow$ Calculate initial vehicle routes

Sort ψ_V descending
while ($\psi_V \neq \emptyset$) do
  $m, n \leftarrow$ Return the nodes of the first element in ψ_V, which has the highest value of ϑ_{mn}
  Get $route_1$ that contains station m, and $route_2$ that contains station n from S_V
  if ($route_1 \neq route_2$ & station m and n are not intermediate nodes) then
    if (m is the first edge in $route_1$ and n is the last edge in $route_2$) then
      $merged \leftarrow$ Merge stations m and n in a new route, which originates from the origin of $route_1$ and ends at the destination of $route_2$.
    end if
    if ($merged \neq \emptyset$) then
      $merged \leftarrow$DistanceImprove ($merged$)
      Remove $route_1$ and $route_2$ from S_V and add $merged$ to S_V
      if ($merged$ is NOT feasible) then
        Remove $merged$ from S_V, and add $route_1$ and $route_2$ to S_V
      end if
    end if
  end if
  Eliminate ϑ_{mn} from ψ_V
end while
Return S_V

Checking the feasibility of the initial solution

The sequence of points in the routes produced at this stage may be inconsistent with the number of onboard drones. In fact, drones may be dispatched from nodes where no drone is available, considering the route taken so far. Figure 3.14 shows an example of this situation.

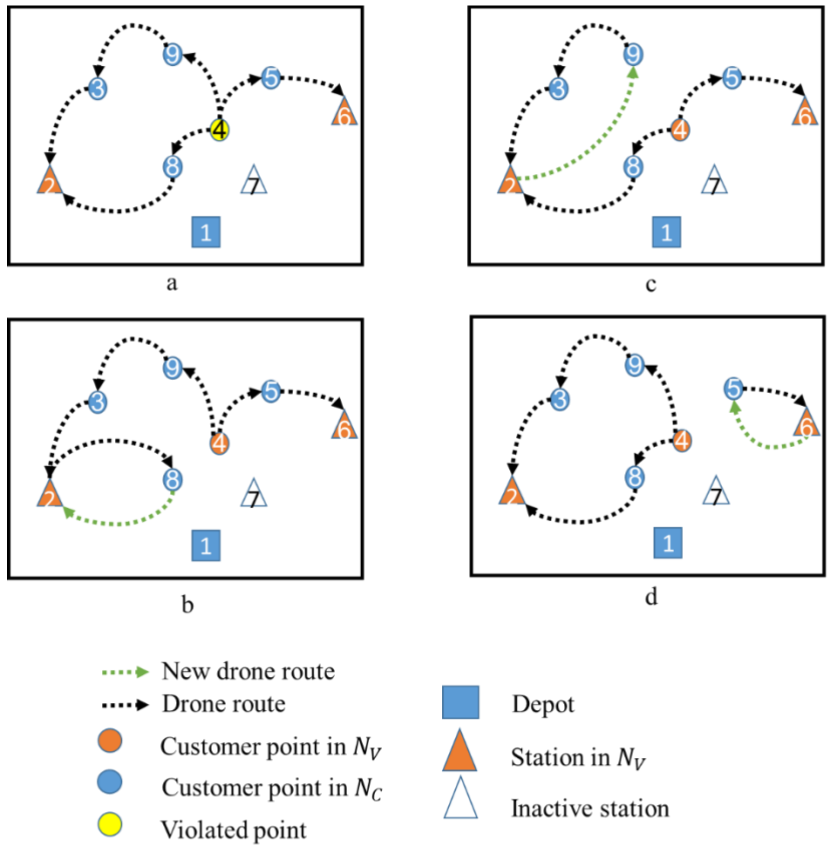


Figure 3.13. The steps to generate a feasible route

At the next step, we use a greedy algorithm to prevent infeasible dispatching and improve the route lengths. H4 represents the steps of the function designed to modify truck routes.

H4: Improve_Vehicle_Routes

```
Input: S_D, sv
Result: sv'
Fv ← the nodes that only dispatch drones regrading to SD.
Lv ← the nodes that only collect drones regrading to SD.
sv' ← the origin of sv
remove the origin of sv
While (all nodes of sv are not included in sv' ) do
    If (Fv ≠ ∅) then
        j ← the closest node to i in sv, which is not in Lv.
    else
        j ← the closest node to i in sv
    End if
    Add j to sv', and remove it from sv.
    If (j ∈ Fv) then
        Remove j from Fv
    End if
End while
Return sv'
```

In H4, for each truck's route, we first specify the points from which the drone is dispatched or collected and store each in a separate list. Then we initiate from the starting point of the route and move to the nearest neighbour node. If the neighbour node is a collecting node and at least one dispatching point has not been visited, we ignore the current neighbour node and move on to the next nearest neighbour. This process continues until we reach the destination point of the truck route. After completing this step, truck and drone routes are created for the problem for given N_V and N_S. Figure 3.14 shows the steps for correcting the truck routes depicted in Figure 3.13.

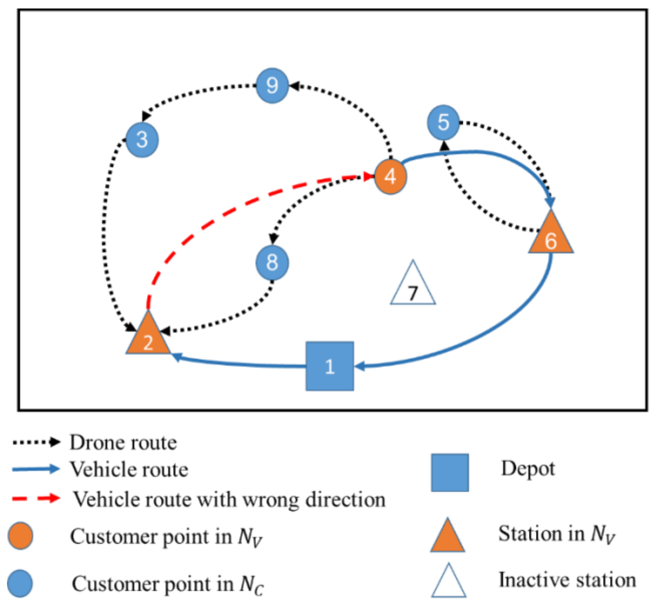


Figure 3.14. An example of the situation where routes produced are inconsistent with the number of onboard drones.

Figure 3.15 (a) shows the nodes to be visited by the truck. In the network displayed, docking station 2 is the closest neighbour to the origin of the truck route. However, since the drone is to be collected at this node, the truck ignores this node and examines the second nearest neighbour. From docking station 6 and customer point 4, docking station 6 is selected due to its proximity to the origin of the truck route, which is depicted in Figure 3.15 (b). The truck then moves on to the next nearest neighbour, which is node 4. Since the drone is not collected from customer point 4, it is feasible for it to be visited by the truck. Figure 3.15 (c) shows this step. Now that all the nodes from which the drone is sent have been visited, docking station 2 is available for the truck to visit. Finally, the modified route is shown in Figure 3.15 (d).

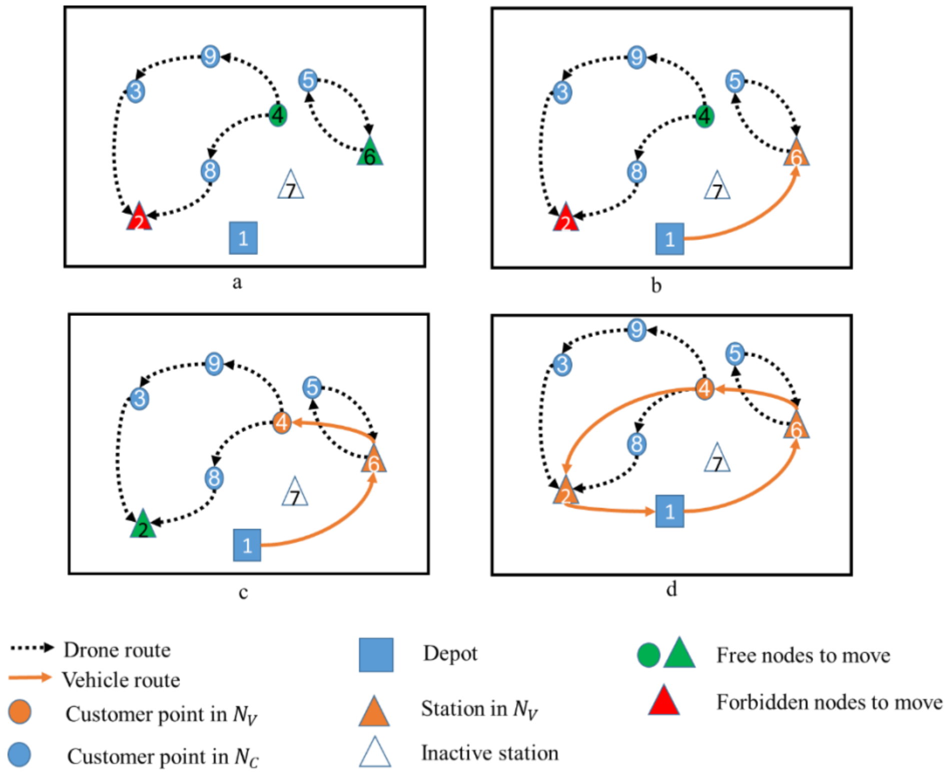


Figure 3.15. The steps for correcting truck routes

3.5.2 Backward and Backward-Forward Clarke and Wright Heuristics: Improving the initial solution

All the steps constitute the main operations of the proposed algorithm. The next step is to decide whether to remove the nodes from the N_V. For this purpose, two different methods with forward and backward operators are considered. In the backward method, nodes are removed one by one from N_V, while in the forward method, nodes are added one by one to N_V . The sequence of these two operators plays a key role in the performance of the algorithm. However, due to the heuristic nature of the backward method it is highly likely that some stations that exist in the optimal solution may be removed. To prevent such occurrences, we develop an enhanced elimination process. If the forward method is used, it must first be ensured that there are no customer nodes in N_V and it is assumed that all customers are visited by drones. This assumption, however, leads to infeasible routes for drones because customers assigned to drones may be outside their flight range or exceed their load carrying capacity. Therefore, the forward method cannot be used first. Besides, due to the heuristic nature of the algorithm, nodes from N_V may be removed during the backward method while in the optimal solution they may exist. Hence, when the termination condition of the algorithm in the backward method is met, the forward method is then applied to add the deleted nodes in the backward step to N_V. This process continues until adding any nodes does not reduce the overall objective function. H7 shows the steps of the Back-and-Forth method.

We show the algorithms in which backward and backward-forward methods are applied with BCWH and BFCWH, respectively.

H7: Back and Forth Clarke and Wright Heuristic (BFCWH)

Input: $N_V, S_V, S_D, BestCost, stage$
Result: $FinalRoutes, BestCost$
$N_V' \leftarrow N_V$
$r \leftarrow \emptyset$, $RmovedNodes \leftarrow \emptyset$
$continue \leftarrow$ true
While ($continue$= true) do
    $continue \leftarrow$ false
    for all ($j \in N_V$) do
        if ($w_j \leq cwd$) then
            Remove j from N_V'
            $\psi_D' \leftarrow$ Considering the N_V', calculate pair saving for drone visited nodes.
            $\psi_V' \leftarrow$ Considering the N_V', calculate pair saving for truck visited nodes.
            $S_D' \leftarrow$ Build_Drone_Routes(N_v',ψ_D')
            $S_D' \leftarrow$ Build_Drone_Routes(S_V',ψ_D')
            According to SD' update the weights of nodes.
            $S_V' \leftarrow$ Build_Vehicle_Routes(S_D', ψ_V')
            $S_V' \leftarrow$ Improve _Vehicle_Routes(S_D', S_V', N_S')
            if (the S_V' and S_D' are feasible) then
                $\delta \leftarrow$ cost ($S_D' \cup S_V'$)
                if ($\delta < BestCost$) then
                    $BestCost \leftarrow \delta$
                    $FinalRoutes \leftarrow S_D' \cup S_V'$
                    $r \leftarrow j$
                    $continue \leftarrow$ true
                end if
            end if
        end if
    $N_V' \leftarrow N_V$
    end for
    Remove r from N_V
    Add r to $RemovedNode$
End while
$continue \leftarrow$ true
$N_V' \leftarrow N_V$
$r' \leftarrow \emptyset$
While ($continue$= true) do
    $continue \leftarrow$false
    for all ($j \in RemovedNode$) do
            add j to N_V'
            $\psi_D' \leftarrow$ Considering the N_V', calculate pair saving for drone visited nodes.

$\psi_V' \leftarrow$ Considering the N_V', calculate pair saving for truck visited nodes.
$S_D' \leftarrow$ Build_Drone_Routes(N_v',ψ_D')
$S_D' \leftarrow$ Build_Drone_Routes(S_V',ψ_D')
According to SD' update the weights of nodes.
$S_V' \leftarrow$ Build_Vehicle_Routes(S_D', ψ_V')
$S_V' \leftarrow$ Improve _Vehicle_Routes(S_D', S_V', N_S')
if (the S_V' and S_D' are feasible) then
$\delta \leftarrow$ cost ($S_D' \cup S_V'$)
if ($\delta < BestCost$) then
$BestCost \leftarrow \delta$
$FinalRoutes \leftarrow S_D' \cup S_V'$
$r' \leftarrow j$
$continue \leftarrow$ true
end if
end if
$N_V' \leftarrow N_V$
end for
Add r' to N_V
Remove r' form $RemovedNode$
End while
return $FinalRoutes, BestCost$

3.6 RESULTS AND ANALYSES

In this section we solve the truck-drone routing problem as represented by our mathematical model for various network realizations using the BFCWH and BCWH heuristics. The truck-drone delivery system is in the early stages of operational deployment and many relevant parameters cannot be chosen precisely. However, every effort has been made make the model as realistic as possible. In this regard, the parameters have been extracted from related literature and their accuracy has been investigated through consultation with subject-matter experts. To avoid biased data, nine different network configurations are considered, for each of which random data for customers' locations and demand is generated.

Table 3.5 summarizes the features of nine networks. The networks under consideration differ in the number of customer points, the number of docking stations, and the level of customer density (customers/km^2). Customer demand for each sample is generated in a range of 1 kg to 15 kg following a uniform distribution so that at least 60% of the demand is portable by drones. The drones are assumed to have a maximum flight range of 2.5 km and a load capacity of 10 kg. Each truck has a load capacity 200 kg and is equipped with two drones.

Table 3.5. Summary of networks used to test the performance of the heuristics

Network	Number of customers	Number of potential stations	Area (mile^2)	Customer density (customers/ mile^2)
A	8	2	1	8
B	10	2	1	10
C	12	3	1	12
D	50	3	1	50
E	50	5	1	50
F	50	10	2	25
G	100	5	1	100
H	100	8	2	50
I	100	10	3	33

The fixed cost per day of the docking stations and trucks is estimated at \$100 and \$100, respectively. The fuel consumption is about 0.15 litter/km ($\approx$0.24 liter/mile) for a truck and oil cost is about \$0.83 per liter (*Gasoline Prices around the World, 02-Nov-2020 | GlobalPetrolPrices.Com*, n.d.). Drone operating cost is set to \$0.05 per km (Kim 2016, n.d.).

All our procedures were implemented in Java (version SE 11) and executed on an x64-PC with an Intel Core i7-7600 1.3 GHz CPU and 16.0 GB of RAM. We used Gurobi version 9.0.2 as the exact solver.

3.6.1 Performance comparison of heuristics for small network instances

Since the execution time required to solve the mathematical model using the standard solver for large-size instances is very long, the heuristics developed are compared with the optimal solutions for small problem sizes only. The performance of BFCWH and BCWH heuristics is reported for several numerical experiments using two measures: The value of the objective function when the program terminates (i.e., converges) and execution time.

Table 3.6 summarizes the performance comparison of BFCWH and BCWH heuristics and the exact solver for a number of small-size problem instances. The results show that both heuristics have a far better performance in terms of execution time compared to the exact solver. The average execution time of both heuristic solvers for all small instances under experiment is less than one second, while the average solving time of the mathematical model for network A and B is 598.9 and 941.7 seconds, respectively. Another noteworthy point is that by adding two customer points to the network A, the solution time increases sharply, showing how a small change in the number of nodes can lead to a considerable increase in problem complexity. However, such significant growth in the solution time is not perceived with a slight increase in the number of nodes for either of the two heuristics. In addition, both heuristics yield quality solutions in terms of optimality by obtaining the optimal solution for some networks tested. The average optimality gap for algorithms BCWH and BFCWH is 1.29% and 1.3% in network A, respectively. The same measure is reported as 1.29% and 1.8% for algorithms BCWH and BFCWH in network B, respectively. Also, as expected, in terms of the average optimality gap, BFCWH outperforms BCWH in all instances analysed.

In order to test the performance of the heuristics more accurately, the mathematical model is fed by the solution obtained from BFCWH to check the extent to which the MIP is able to improve the heuristic solution after a maximum run-time of 1800 seconds. For this purpose, five random datasets are generated according to Network C and the execution time of the MIP is limited to 1800 seconds. As can be observed in Table 3.6, on average, after 1723.2 seconds, the mathematical model is only able to improve the initial BFCWH solution by 2.8%. Note that 1723.2 seconds of run-time is not the time it takes for the MIP to solve the problem in Network C, but the amount of time it spends improving the BFCWH solution. When the runtime hits 1800 seconds for the MIP solver, some solutions have not converged.

Table 3.6. Performance comparison between the exact solution and heuristics.

Instance	MIP solution			BFCWH			BCWH		
	T (sec)	Cost ($)	Gap	T (sec)	Cost ($)	Gap	T (sec)	Cost ($)	Gap
A1	1201	652.022	0%	<1	681.08	4.4%	<1	681.08	4.4%
A2	620	631.56	0%	<1	631.56	0%	<1	631.56	0%
A3	1132	534.479	0%	<1	544.01	1.7%	<1	544.48	1.8%
A4	47	739.5	0%	<1	749.50	1.3%	<1	749.50	1.3%
A5	457	696.27	0%	<1	696.27	0%	<1	696.27	0%
A6	1200	659.74	0%	<1	675.62	2.4%	<1	675.62	2.4%
A7	614	716.5	0%	<1	716.5	0%	<1	716.5	0%
A8	122	570.41	0%	<1	586.37	2.8%	<1	586.37	2.8%
A9	570	628.02	0%	<1	628.02	0%	<1	628.02	0%
A10	26	656.856	0%	<1	658.99	0.3%	<1	658.99	0.3%
A mean	598.9	648.53	0%	<1	656.79	1.29%	<1	660.93	1.3%
B1	789	777.98	0%	<1	787.99	1.3%	<1	787.99	1.3%

B2	1200	742.49	0%	<1	742.49	0%	<1	742.49	0%
B3	707	832.50	0%	<1	847.30	1.7%	<1	859.55	3.2%
B4	1200	846.91	0%	<1	856.92	1.2%	<1	856.92	1.2%
B5	1120	690.15	0%	<1	706.79	2.4%	<1	701.48	1.6%
B6	840	726.73	0%	<1	742.70	2.2%	<1	758.48	4.3%
B7	780	698.7	0%	<1	711.14	1.7%	<1	721.59	3.2%
B8	509	633.02	0%	<1	646.71	2.1%	<1	651.41	2.9%
B9	1342	695.08	0%	<1	697.35	0.3%	<1	697.35	0.3%
B10	930	629.19	0%	<1	629.19	0%	<1	632.02	0.45%
B mean	941.7	727.27	0%	<1	736.85	1.29%	<1	740.92	1.8%
C1	1800	662.44	25.9%	<1	683.70	3.2%	<1	683.70	3.2%
C2	1800	642.55	10.3%	<1	661.34	2.9%	<1	657.04	2.2%
C3	1536	652.02	0%	<1	662.02	1.5%	<1	662.02	1.5%
C4	1680	635.36	0%	<1	658.55	3.6%	<1	664.27	4.5%
C5	1800	684.06	10.5%	<1	703.3	2.8%	<1	711.93	4%
C mean	1723.2	655.28	9.3%	<1	673.78	2.8%	<1	675.79	3%

3.6.2 Performance comparison for large-size network instances

The performance of BFCWH and BCWH heuristics is reported for large problem sizes in this section. The two algorithms are compared in terms of execution time and total cost, and the results are discussed. Tables 3.7 and 3.8 show the results of the performance of these heuristics with random data described earlier consisting of 50 customers in networks D, E and F; and 100 customers in networks G, H, and I, respectively. The configuration of each network is given in Table 3.5. The heuristics are analysed ten times with random data at each network instance. In each test, the total cost (C_{total}), operating cost of trucks (C_{Tr}), operating cost of drones (C_D), the number of active stations (N_S), the number of trucks (N_{Tr}), and the number of drones (N_D) in operation are reported. Two examples of truck-drone cooperative networks produced by the FBCWH and BCWH algorithms are shown in Figure 3.16 and 3.17 respectively.

Depots are depicted as blue squares, docking stations as triangles, customer locations as circles, truck routes as solid lines, and drone routes as dashed lines (different colours applied for different vehicles).

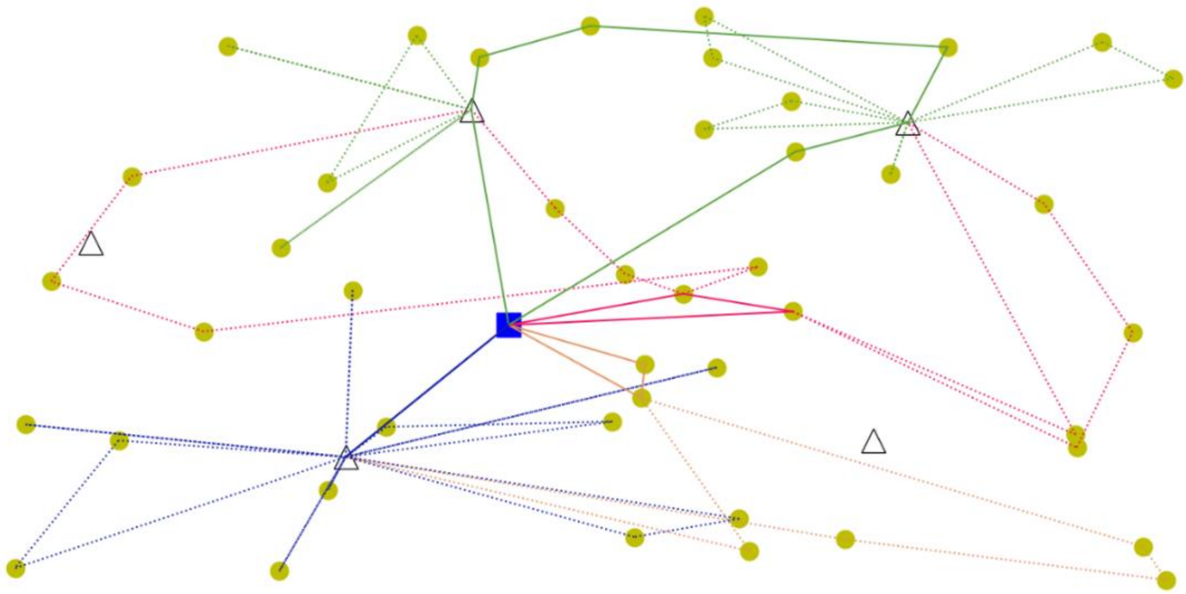


Figure 3.16. Truck-drone network with flexible mothership obtained from running BCWH (Total cost =2029.96)

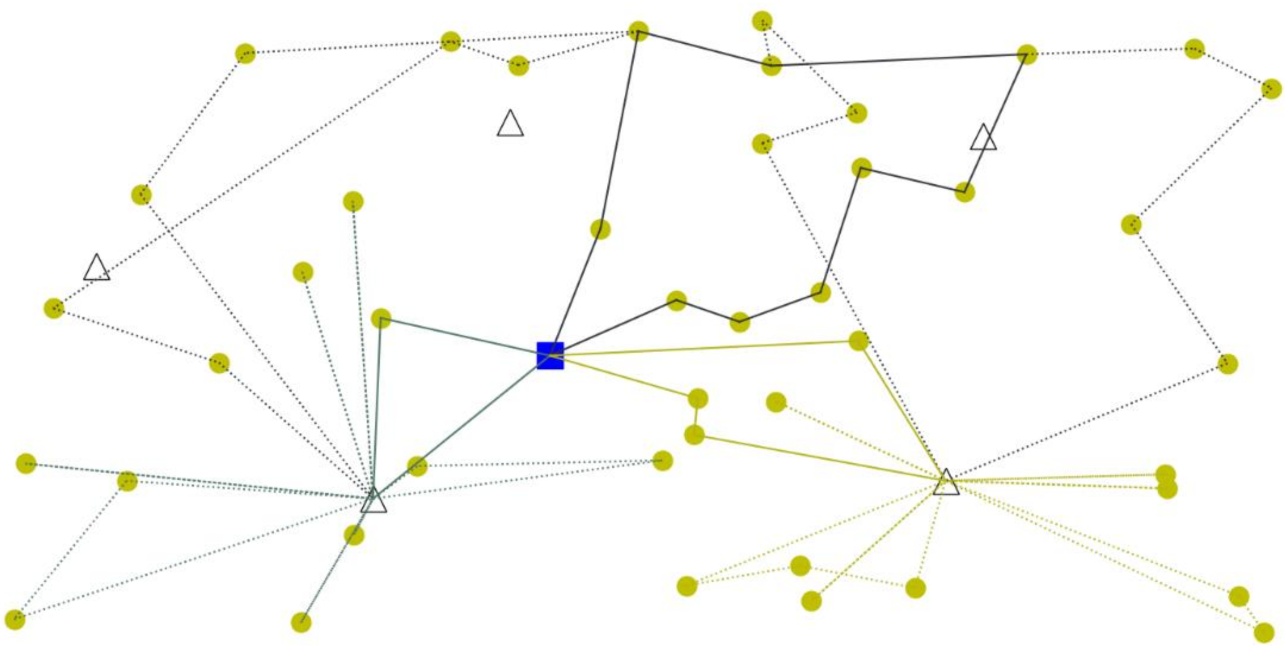


Figure 3.17. Truck-drone route network with flexible mothership obtained from FBCWH (Total cost=1927.92)

From Tables 3.7 and 3.8 it may be observed that BFCWH usually produces better solutions than BCWH in terms of the total cost. Notice that BFCWH is typically 20% slower than BCWH. This is because of the additional steps required to determine which nodes need to be removed from the network to improve solution quality.

Table 3.7. The performance of our two new heuristics for 50 customers.

Ins.	BFCWH							BCWH						
	T	C_{total}	C_D	C_{Tr}	N_D	N_{Tr}	N_S	T	C_{total}	C_D	C_{Tr}	N_D	N_{Tr}	N_S
D1	17	2110.52	184.16	1746.36	2	3	1	13	2247.99	586.09	1341.90	4	4	2
D2	19	2146.67	420.53	1426.14	4	3	2	14	2234.35	252.79	1681.56	4	3	2
D3	19	2557.87	548.09	1569.78	6	4	3	12	2391.49	377.53	1713.96	4	3	2
D4	14	2295.77	216.05	1899.72	2	3	1	12	2463.19	117.95	2055.24	3	3	2
D5	16	2351.34	244.50	1806.84	4	3	2	12	2374.91	230.27	1844.64	4	3	2
D6	20	2078.98	411.16	1367.82	4	3	2	13	2160.37	664.79	1175.58	4	4	2
D7	15	2165.73	436.33	1409.40	4	4	2	13	2323.65	392.29	1611.36	4	4	2
D8	14	2100.73	301.69	1499.04	4	3	2	13	2121.28	477.76	1343.52	4	3	2
D9	15	2348.90	301.74	1757.16	3	3	2	16	2441.60	179.14	1862.46	4	3	3
D10	17	2254.28	420.12	1514.16	4	4	2	13.	2318.60	659.82	1218.78	6	4	3
Mean	17	2241.07	348.44	1599.64	3.7	3.3	1.9	13	2307.74	393.84	1584.9	4.1	3.4	2
E1	24	2583.51	248.33	2115.18	2	5	1	18	2640.97	390.45	1910.52	4	5	2
E2	24	2382.50	230.84	1851.66	4	3	2	19	2572.16	413.46	1838.70	4	4	2
E3	22	2563.44	308.10	1955.34	4	3	2	18	2544.09	283.89	1960.20	4	3	2
E4	26	2347.37	268.07	1779.30	4	3	2	23	2461.61	191.31	1860.30	5	3	3
E5	28	2440.56	347.22	1793.34	4	3	2	23	2786.99	367.63	1989.36	5	4	3
E6	25	2561.97	449.73	1812.24	4	3	2	22	2663.40	528.62	1704.78	5	4	3
E7	28	2533.87	258.53	1955.34	4	4	2	16	2497.59	284.35	1893.24	4	4	2
E8	26	2434.95	235.35	2019.60	2	3	1	19	2525.65	199.69	2145.96	2	3	1
E9	25	2224.75	348.07	1696.68	2	3	1	17	2459.35	403.79	1735.56	4	4	2
E10	26	2855.66	206.10	2329.56	4	4	2	18	2881.86	112.42	2449.44	4	4	2
Mean	25	2492.86	290.03	1930.82	3.4	3.4	1.7	19	2603.37	317.56	1948.8	4.1	3.8	2
F1	11	5031.01	351.03	4069.98	5	3	5	93	5109.16	407.38	3891.78	5	3	7
F2	12	4454.24	303.12	3741.12	5	3	3	97	5619.15	375.51	4193.64	7	4	9
F3	10	4776.85	326.65	3850.20	4	3	5	119	4410.27	508.25	3382.02	6	3	4
F4	12	4219.49	344.59	3474.90	4	3	3	136	4401.77	298.09	3613.68	3	3	4
F5	11	4599.37	460.19	3519.18	6	3	5	138	4488.19	415.53	3552.66	6	3	4
F6	51	4828.41	242.33	4186.08	4	3	3	130	4888.95	683.43	3665.52	6	4	4
F7	12	5161.69	160.43	4411.26	3	3	5	135	4967.47	243.49	4123.98	4	3	5
F8	13	4560.91	413.31	3747.60	4	3	3	144	4416.34	487.00	3629.34	4	3	2
F9	96	4817.45	357.43	3760.02	4	3	6	123	4617.62	416.34	3581.28	6	3	5
F10	60	5462.05	286.07	4555.98	4	4	5	96	5811.96	566.36	4395.60	7	4	7
Mean	10	4791.15	324.91	3931.63	4.3	3.1	4.3	121	4873.1	440.14	3802.95	5.4	3.3	5

From Tables 3.7 and 3.8 show that the average operational cost of drones in the BFCWH method is higher than the BCWH method, indicating that the BFCWH algorithm services more customers using drones. The average number of drones required by BFCWH is lower than BCWH and it is apparent that BFCWH generates longer drone routes, whilst reducing the number of drones needed to service customer demand.

As seen in Tables 3.7 and 3.8, the number of trucks and the operating cost of trucks in BFCWH are lower than those for BCWH. The main reason for this is that in BFCWH, more customers are served by drones, and consequently, the number of trucks required will be lower, and their routes shorter than those for BCWH. Shorter truck routes and the more efficient use of drones means that fewer stations are required in the network, as may be seen in Figures 3.16 and 3.17.

Table 3.8. The performance of our two new heuristics for 100 customers.

	BFCWH							BCWH						
Ins.	T	C_{total}	C_D	C_{Tr}	N_D	N_{Tr}	N_S	T	C_{total}	C_D	C_{Tr}	N_D	N_{Tr}	N_S
G1	433	4144.15	289.59	3274.56	6	6	4	625	3760.76	288.22	3132.54	4	5	2
G2	548	3801.42	368.04	2833.38	8	6	4	565	3703.73	218.11	3025.62	6	5	3
G3	426	4177.54	295.44	3302.1	6	6	4	546	3980.07	205.55	3314.52	4	6	3
G4	540	3839.83	286.29	3213.54	4	5	2	470	4071.72	399.94	3081.78	7	6	4
G5	554	4006.66	271.46	3175.2	4	6	4	605	3728.92	261.22	3107.7	4	6	2
G6	589	4127.39	467.21	3060.18	8	6	4	511	4255.91	468.57	3197.34	7	6	4
G7	628	3908.99	172.55	3286.44	5	5	3	622	3952.49	198.77	3303.72	5	5	3
G8	782	3714.06	270.96	3113.1	3	5	2	642	3871.97	433.87	2978.1	4	6	3
G9	586	3900.44	210.3	3100	7	6	4	597	3871.97	433.87	2978.1	4	6	3
G10	674	3474.84	157.28	2977.56	4	5	2	516	3874.57	214.85	2979.72	8	5	5
Mean	576	3909.53	278.912	3133.60	5.5	5.6	3.3	569.9	3907.211	312.29	3109.91	5.3	5.6	3
H1	753	7045.47	864.39	5401.08	6	6	6	703	7310.79	328.93	6241.86	4	5	6
H2	1212	6481.55	408.47	5293.08	6	6	4	1116	6546.67	669.29	5317.38	4	6	4
H3	754	6363.35	823.87	4839.48	8	6	5	939	6533.65	437.73	5425.92	7	5	5
H4	1191	6396.12	526.04	5320.08	5	5	4	699	6758.63	677.71	5290.92	9	5	6
H5	1210	6257.4	467.08	5350.32	4	5	3	951	6480.44	508.48	5331.96	4	5	5
H6	1179	6745.33	457.09	5808.24	6	6	3	1061	6909.88	429.56	5890.32	7	6	4
H7	728	6975.36	823.52	5451.84	8	6	5	1091	7084.25	846.39	5647.86	7	6	4
H8	713	6854.02	663.42	5610.6	6	6	4	923	6840.57	720.12	5429.7	7	6	5
H9	801	6531.34	975.14	4876.2	6	6	5	720	6591.12	990.14	4629.7	7	6	5
H10	760	6632.2	40.36	5775.84	5	5	3	954	6632.2	406.36	5775.84	5	5	3
Mean	930.1	6628.14	604.938	5372.67	6	5.7	4.2	915.7	6768.82	601.47	5498.14	6.1	5.5	5
I1	1255	9983.48	1058.22	8245.26	6	6	5	1760	9952.03	997.63	8294.4	4	6	5
I2	1389	9257.13	1189.23	7227.9	12	6	6	1908	9257.13	1189.2	7227.9	12	6	6
I3	1259	9604.71	488.03	8446.68	7	5	5	1474	9953.47	700.03	8443.44	9	6	6
I4	1601	10033	1290.78	8042.22	8	6	5	1689	9797.07	476.27	8650.8	7	5	5
I5	981	9369.35	697.29	8310.06	9	5	5	1705	9797.07	476.27	8650.8	7	5	5
I6	1616	9381.61	1162.57	7439.04	6	6	6	1598	9483.96	724.54	7869.42	9	5	7
I7	1480	10277.6	484.05	8783.64	9	6	8	1011	10277.69	484.05	8783.64	9	6	8
I8	976	10465.8	1664.53	7861.63	10	7	7	896	10356.66	1042.2	8314.38	8	6	8
I9	894	9465.85	1164.53	8361.63	7	7	7	741	95346.46	982.28	84929.8	7	6	8
I10	1001	9512.05	842.63	7869.42	10	5	6	1686	9065.35	960.63	7434.72	7	5	5

Mean	1245	9735.07	1004.18	8058.74	8.4	5.9	6	1447	18328.68	803.32	15859.9	7.9	5.6	6

3.6.3 Flexible mothership versus inflexible mothership and truck-only operation

Having a flexible mothership means that drones can rejoin any nearby truck which is not necessarily its mother truck. This section examines potential cost savings when considering flexible mothership-truck-drone operations. For this purpose, random data is generated for networks D, E, F, G, H, and I, and potential cost savings compared to inflexible mothership and truck-only operations are examined.

Also, flexible and inflexible mothership operations are analysed by considering different flight ranges for drones and different maximum drone-carrying capacity for trucks ($Maxd$). The maximum drones' flight range—between 1 and 3 km—and the maximum drone carrying capacity for trucks, ranging between 1 and 3, are analysed. All numerical results reported in Table 3.9 are obtained from running the BFCWH algorithm. In truck-only operations, shown as VRP in Table 3.9, the drone-carrying capacity of trucks is set to zero. As reported in Table 3.9, in all numerical tests under the same parameters, the truck-drone operation with flexible mothership is the most cost-effective.

As can be observed in Table 3.9, total cost decreases as drone flight range increases. This cost reduction occurs because more customers are serviced by drones, which have lower operating costs than trucks.

For example, in the network I2 with flexible mothership operation and a truck drone-carrying capacity of 2, with an increase in flight range from 1 km to 3 km, approximately 4.8% cost saving is achieved.

Table 3.9. Numerical experiments for flexible mothership, inflexible mothership, and truck-only operations

		Flexible Mothership					Inflexible Mothership					VRP
		Flight range (km)					Flight range (km)					
Ins.	$Maxd$	1	1.5	2	2.5	3	1	1.5	2	2.5	3	
D1	1	2485.82	2434.58	2423.43	2369.27	2352.5	2564.52	2549.33	2510.83	2494.32	2473.71	
D2	2	2370.58	2314.17	2297.98	2281.43	2276.8	2463.63	2426.27	2387.44	2416.42	2384.82	2833.12
D3	3	2318.04	2275.57	2234.76	2179.99	2170.19	2441.14	2409.62	2312.74	2263.52	2242.44	
E1	1	2708.61	2678.03	2625.42	2566.65	2560.82	2837.8	2788.13	2743.51	2687.97	2637.95	
E2	2	2599.52	2560.83	2549.25	2492.86	2488.52	2726.04	2636.98	2669.71	2576.82	2598.71	2912.25
E3	3	2644.5	2602.57	2580.31	2531.39	2516.39	2756.68	2690.9	2708.38	2666.67	2594.16	
F1	1	3521.44	3493.23	3452.01	3386.44	3371.75	3618.74	3594.74	3556.09	3470.09	3487.46	6223.45
F2	2	4871.98	4849.66	4824.04	4791.15	4773.25	4960.35	4960.39	4941.57	4886.54	4880.05	

F3	3	3605.74	3579.49	3545.91	3520.59	3503.2	3721.67	3663.63	3664.2	3613.79	3616.2	
G1	1	4240.85	4200.22	4183.56	4143.42	4126.94	4357.79	4307.17	4302.46	4294.55	4251.22	
G2	2	4020.06	3954.47	3935.61	3912.53	3885.92	4101.01	4063.12	4099.71	4044.56	3993.84	4994.18
G3	3	3963.66	3925.92	3905.26	3879.35	3859.81	4047.82	4090.13	4029.48	3949.36	3943.83	
H1	1	6960.99	6891.66	6848.53	6761.28	6724.32	7089.1	7002.85	7037.45	6867.8	6810.42	
H2	2	6833.02	6747.94	6710.19	6628.14	6613.12	6963.08	6909.47	6880.29	6788.5	6729.9	7987.90
H3	3	6703.31	6652.89	6604.2	6519.37	6474.67	6902.62	6786.24	6765.6	6606.09	6613.72	
I1	1	10055.48	9967.41	9929.77	9876.48	9857.32	10234.29	10106.52	10057.3	10040.5	10020.1	
I2	2	9942.11	9866.81	9808.65	9735.07	9717.33	10120.03	10038.8	9978.74	9830.15	9894.29	13456.74
I3	3	9595.2	9593.89	9587.93	9584.13	9581.09	9607.99	9603.76	9604.49	9596.79	9589.86	

Furthermore, by increasing the drone flight range to 3 km in truck-drone operation with flexible mothership in network I2, approximately 1.7% and 28% cost savings are achieved, compared to inflexible mothership and truck-only operations respectively. Figures 3.18, 3.19, and 3.20 show the three operating methods of truck-drone with flexible mothership, inflexible mothership, and truck-only for a network with the same configuration, respectively.

3.6.4 Trade-off between drone flight range and load carrying capacity

Figure 3.21 shows trade-off between the drone flight range and load carrying capacity for different network configurations. A noteworthy point is the ratio for which the lowest objective function value occurs for networks. In network I, which has larger area with denser population compared to other networks, this ratio is equal to 2km^2 / 13kg. As the service area and number of customers decreases, drones with shorter flight range and greater capacity become more desirable since their associated costs decrease. More carrying capacity for drones leads to a lower cost, while flight lengths become more important for large networks. It is also concluded that for networks with higher customer density, drones with greater capacity are more desirable, while for sparser networks, flight range is more important.

3. CONCLUSION

After an extensive literature review, this chapter presented an MILP mathematical model of a delivery system using trucks and drones. The mathematical model is composed of truck and drone operational parameters and customer locations. To provide a comprehensive decision model, all truck-drone collection-dispatch strategies are considered, including dispatch-waiting-collection or dispatch-transfer-collection scenarios. In this system, items may be delivered to customers either by truck or drone, the assignment being carried out as part of the solution process. The drones are dispatched from trucks to deliver orders and fly toward customer locations. Drones must land at special stations, called docking stations, to join the trucks.

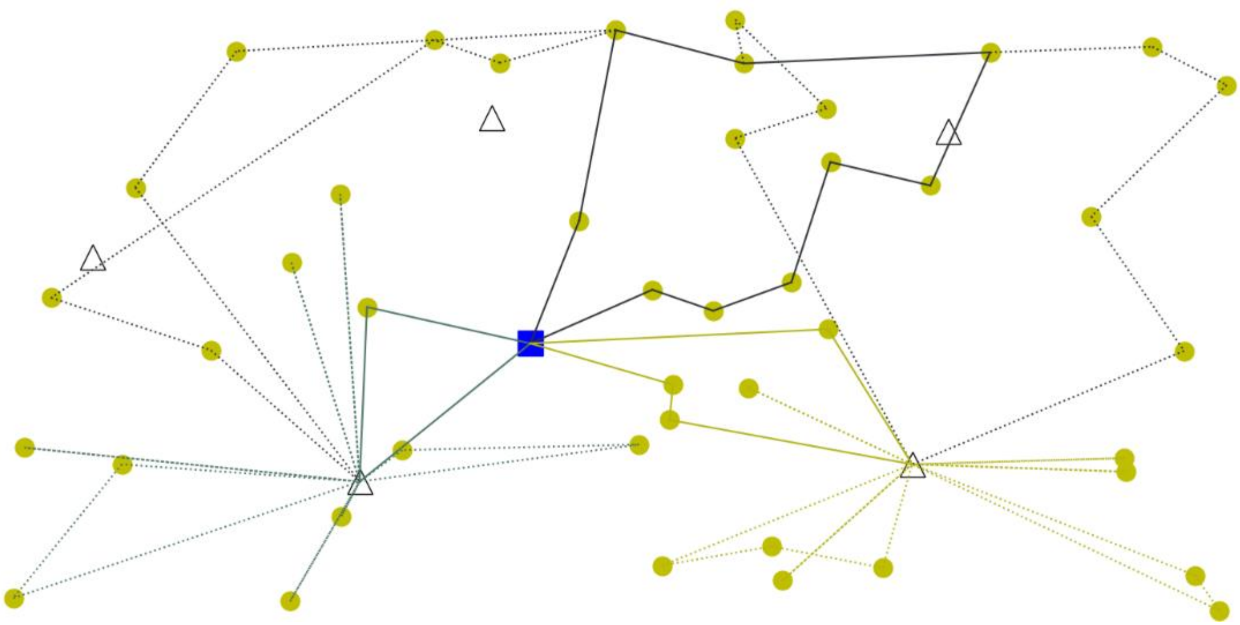


Figure 3.18. Truck-drone operation with flexible mothership (Total cost=1865.30)

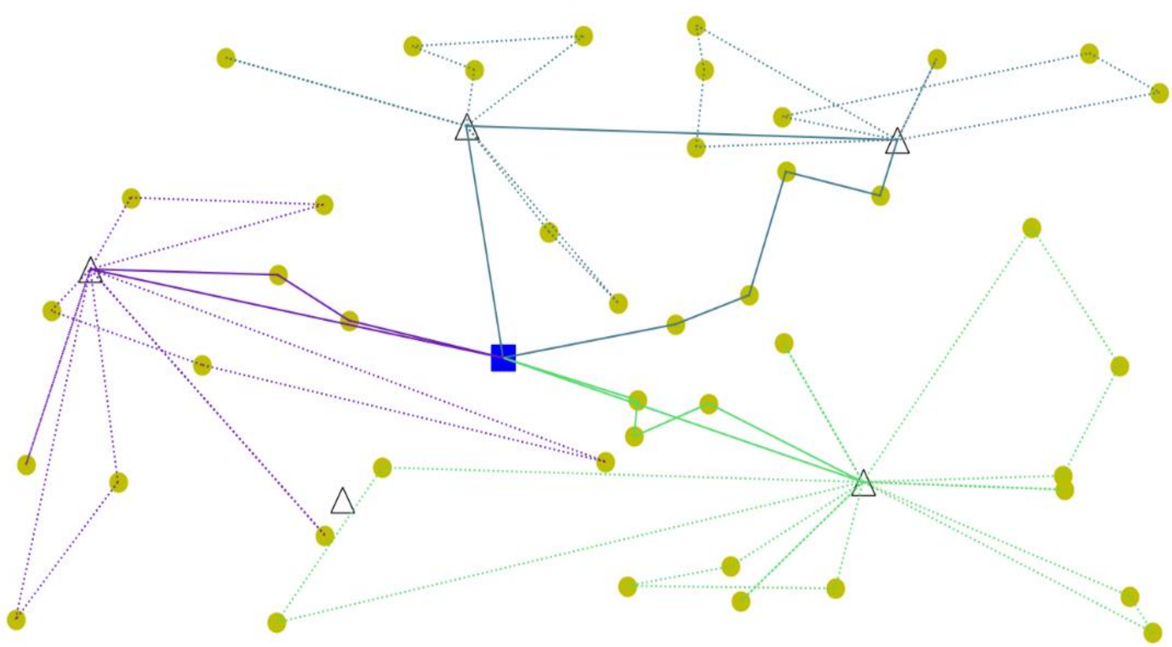


Figure 3.19. Truck-drone operation with inflexible mothership (Objective=2101.85)

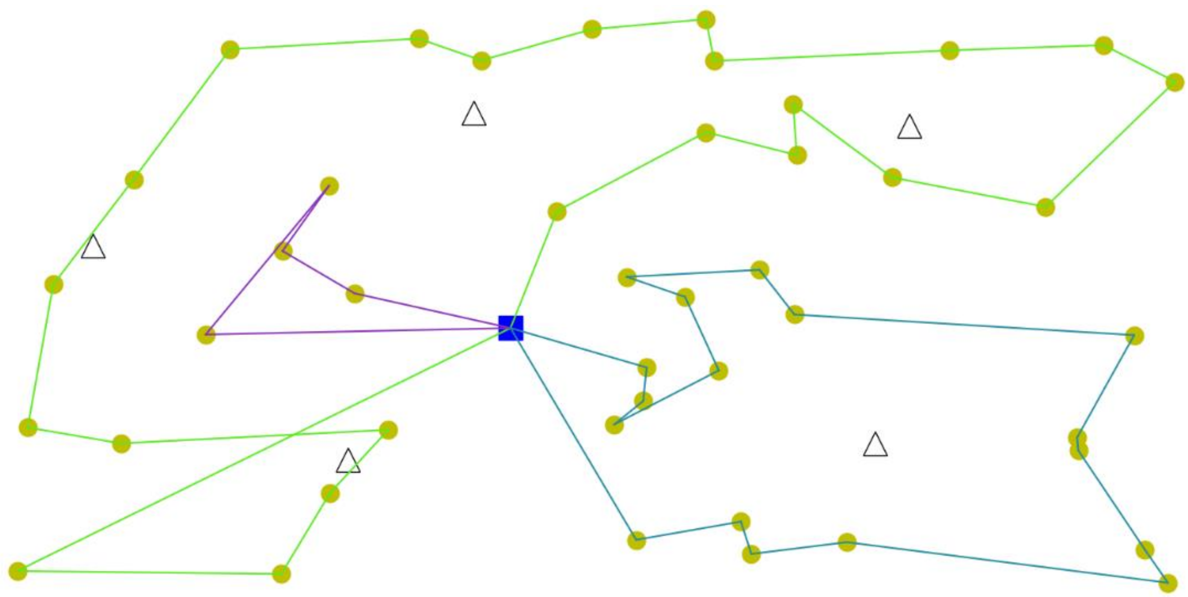


Figure 3.20. Truck-only operation (Objective=2738.60)

Drones can join any truck, not just the truck from which they were dispatched. This flexibility in truck usage results in significant savings which was demonstrated in the numerical experiments.

Since the problem of vehicle routing is intractable (takes too long to solve to optimality with conventional solvers), two efficient algorithms using a combination of heuristics were developed to solve the problem for large problem instances, the efficiency of which was also tested by experiments.

The two proposed heuristics are built upon the Clarke and Wright Algorithm, which is utilized to construct the initial solution. We introduced two distinct methods, incorporating forward and backward operators, aimed at enhancing the initial solution. Our results demonstrate that the Backward-Forward operator (BFCWH) outperforms the Backward operator (BCWH) in terms of solution quality. However, BCWH exhibits a relatively faster convergence to a solution.

The results showed good performance for the heuristic algorithms, both in terms of execution time and optimality.

Sensitivity analysis was also performed on several parameters, which showed that in networks with different configurations and different population densities, the optimal ratio of flight range and load carrying capacity of drone can be different. It was concluded that for more densely populated networks, a higher carrying capacity for drones is more desirable, while for larger networks, flight range becomes more important.

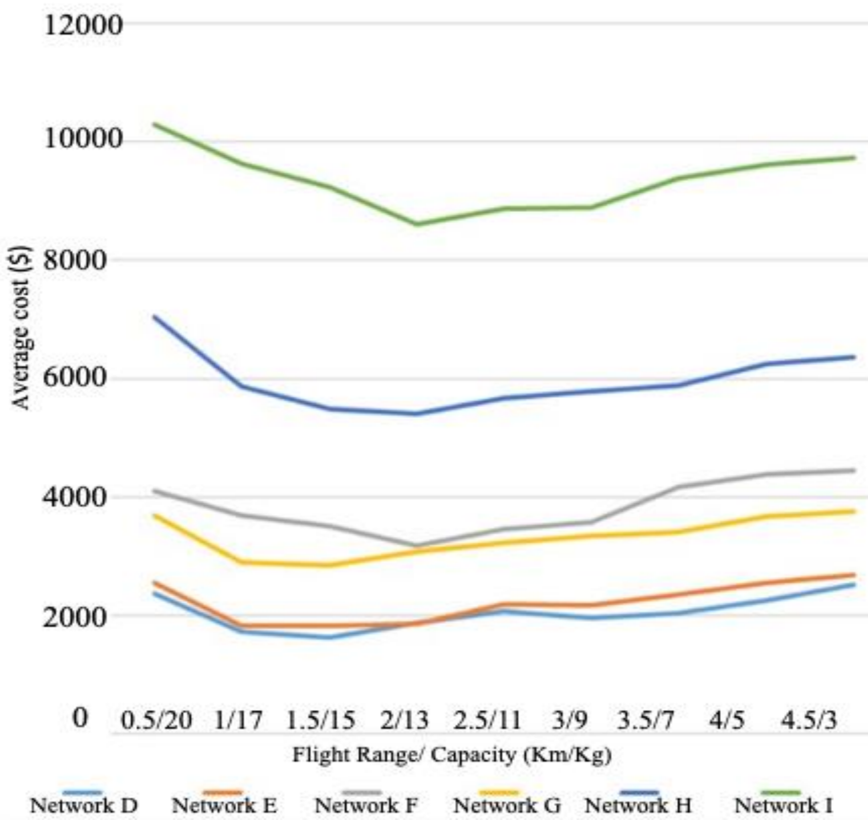


Figure 3.21. Flight range versus load capacity

Future research could involve incorporating scheduling decisions into the formulation of the problem. More efficient algorithms can also be developed to provide quality solution with less execution time.

Chapter 4 introduces an innovative Mixed Integer Programming (MIP) model designed specifically for the delivery of e-groceries to customers' residences. Referred to as the VRPDMTW, this model is an extension of the VRPD, allowing customers to select multiple time windows for delivery. The proposed approach involves a combination of truck and drone deliveries, incorporating synchronization constraints at rendezvous points. The routing strategy entails dispatching trucks equipped with drones to customer locations, followed by subsequent stops for either customer service or drone retrieval.

Chapter 4: Optimizing Truck-enabled Drone Routing with Multiple Time Windows for E-groceries

4.1 INTRODUCTION

Supermarkets are increasingly turning to web-based commerce to keep up with the growing trend of online shopping and meet customer demands for faster delivery. This chapter explores a delivery system that utilizes a network consisting of a depot and a fleet of drones and trucks to deliver items to customers within a short timeframe. In this delivery network, trucks equipped with multiple drones serve as mobile depots for the drones, though the drones can also operate directly from the depot. Delivery is carried out collaboratively by both trucks and drones, with multiple non-overlapping time windows available for customers to choose from. It is important to note that meeting the delivery time window is a strict requirement.

The preferred delivery times for customers are influenced by several factors. External factors such as the time of day, day of the week, and location play a role in shaping customers' preferences. For instance, customers may prefer deliveries during evenings or weekends when they are more likely to be at home. In urban areas, customers might opt for off-peak hours to avoid traffic congestion. Internal factors, including work schedules, family commitments, and personal preferences, also impact preferred delivery times. For example, customers with young children may prefer deliveries during nap times, while those who work from home may prefer deliveries during working hours. E-grocery home delivery companies should take these factors into consideration when offering delivery time options to their customers. By utilizing data analytics, companies can gain insights into their customers' preferences and provide tailored delivery options that align with their customers' needs.

In the realm of e-commerce and online retail, traditional delivery methods using only trucks often struggle to meet the short delivery windows demanded by e-grocery customers.

This study aims to explore the Vehicle Routing Problem with Multiple Time Windows (VRPMTW) and its optimal solution that involves both truck and drone routes, known as the Vehicle Routing Problem with Drone considering Multiple Time Windows (VRPDMTW). If customers select multiple time windows according to their convenience, the proposed system assigns the option that results in the lowest cost. However, this study does not encompass the discounted delivery fee offered to customers who choose multiple delivery windows. To provide a visual representation, Figure 4.1 depicts a small instance of the hybrid home delivery network.

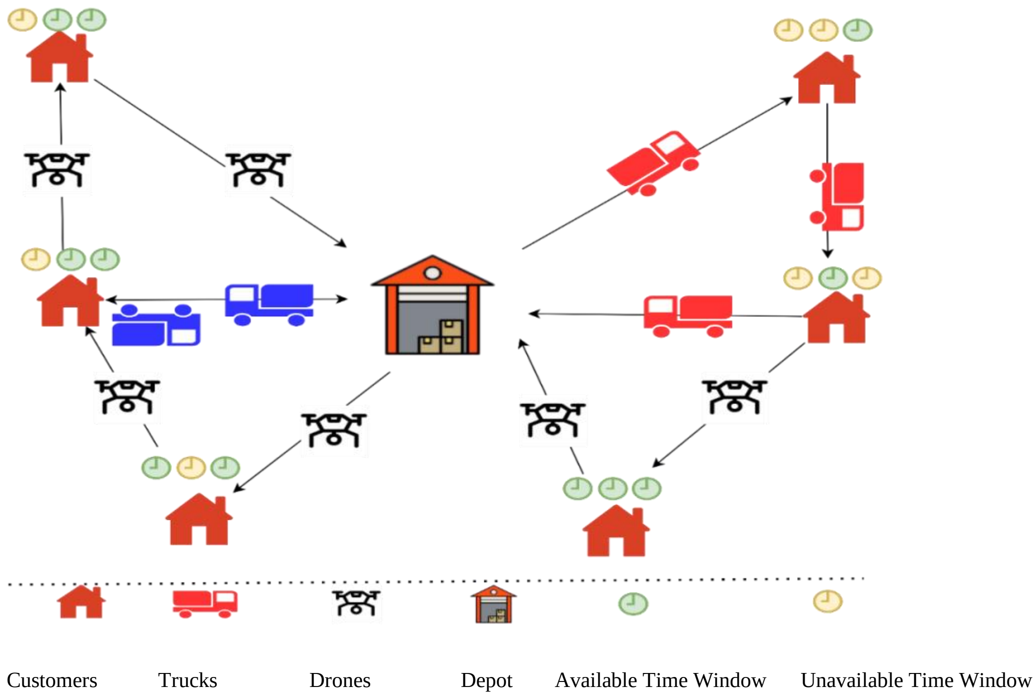


Figure 4.1. An example of drone-truck network with multiple time windows

In this problem, customers may have one or multiple time windows and require a specific service time, which can vary for trucks and drones. Each customer has a location and an order with a known size, while the trucks have limited capacity and start and end their route at the depot. The cost and time of travel vary between all pairs of locations for both trucks and drones. Drones are deployed at customer locations and must be retrieved at a different customer location by the same truck, or they can return directly to the depot. However, drones have limited flying time, and each sortie serves only one customer. The study assumes that drones cannot wait at customer locations and must be picked up by the truck immediately upon arrival.

4.1.1 Contributions of this study

The main contributions of this chapter are as follow:

- This study considers the possibility of choosing multiple time windows for customers, which offers a more generalized drone-truck routing problem with time windows.
- This study presents a Constraint Programming model to solve the sub-problem of finding the optimum schedule for a given truck/drone route exactly.
- This study proposes an Adaptive Large Neighbourhood Search (ALNS) heuristic to deal with the complexity of the drone-truck routing problem for large instances.

The remainder of this chapter is organized as follows. In Section 4.2 the literature review is presented. The mathematical formulation is presented in Section 4.3. Section 4.4 presents the solution methodology. In Section 4.5, the numerical experiments to examine the algorithm performance are presented and sensitivity analysis on the key parameters is performed. Finally, the conclusion is provided in Section 4.6.

4.2 REVIEW OF THE RELEVANT WORKS

Murray & Chu (2015) introduced the Flying Sidekick Traveling Salesman Problem (FSTSP) and the Parallel Drone Scheduling TSP (PDSTSP) to optimize parcel deliveries. The FSTSP involves a truck acting as a mobile depot for a drone, while the PDSTSP aims to optimize parcel deliveries with non-cooperative drone and truck operations. Ham (2018) extended the FSTSP to allow drones to pick up and deliver items independently. Sacramento et al. (2019) formulated the FSTSP as a Mixed-Integer Programming model to minimize cost and proposed an Adaptive Large Neighbourhood Search algorithm to solve the model for large networks. Table 4.1 compares this study with the most relevant studies in the literature.

Agatz et al. (2018) introduced the Traveling Salesman Problem with Drone (TSP-D), an extension of the FSTSP where a drone can serve a set of customers on each mission. In TSP-D, the truck serves as a mobile depot while the drone is dispatched to docking stations for product delivery. Ha et al. (2018) aimed to minimize operational costs in a TSP-D and proposed two algorithms, a local search, and a Greedy Randomized Adaptive Search Procedure (GRASP), to handle large instances. Karak & Abdelghany (2019) developed a MIP model for a truck-drone system where the drone delivers items to end customers and the truck serves as a mobile depot. Roberti & Ruthmair (2021) addressed various TSP-D variants and proposed a compact mixed-integer linear program (MILP) that considers synchronization constraints between the truck and drone. Finally, Mara et al. (2022) developed an Adaptive Large Neighbourhood Search (ALNS) algorithm and a mathematical model for the Flying Sidekick Traveling Salesman Problem (FSTSP) where drones can visit multiple points within their routes.

Several subsequent studies have attempted to refine and expand upon the assumptions made in the FSTP. The Vehicle Routing Problem with Drones (VRP-D) has been studied extensively, with various approaches taken to optimize delivery processes. (Boysen, Briskorn, et al., 2018) optimized the scheduling of drone deliveries within a predetermined truck route, with drones waiting for the truck to arrive. Kitjacharoenchai et al. (2019) aimed to minimize delivery time by allowing drones to land on nearby trucks but did not consider truck capacity. (Schermer et al. (2019) used a MILP model to minimize makespan for a fleet of trucks with drones, developing a metaheuristic algorithm to deal with large instances. Wang & Sheu (2019) allowed trucks to move to subsequent stations to collect drones, while Kitjacharoenchai et al. (2020) minimized total arrival time of trucks and drones at the depot by using a truck as a mobile intermediate depot equipped with multiple drones.

The Multi-visit Vehicle-Drone Routing Problem (MVRP-D) extends the Vehicle Routing Problem with Drones (VRP-D) by assuming drones have enough capacity to visit multiple customers per mission. Luo et al. (2021) formulated MVRP-D with multi-drones as a Mixed Integer Programming (MIP) model to minimize makespan. Despite the importance of time window constraints in last-mile delivery, few studies have considered the drone-truck routing problem with time windows. Di Puglia Pugliese & Guerriero (2017) addressed the drone-truck routing problem with time windows (VRPTWD) for a single truck and drone, while Ham (2018) extended the problem to multiple trucks and drones. Cheng et al. (2020) considered the drone routing problem with time windows and developed a branch-and-cut algorithm, while Di Puglia Pugliese et al. (2021) formulated the VRPTWD as a MIP model with a two-phase heuristic. Coindreau et al. (2021) and Kuo et al. (2022) also addressed the VRPTWD as a MIP model and developed Adaptive Large Neighborhood Search and Variable Neighborhood Search algorithms, respectively. However, their assumptions regarding negligible drone launching and retrieval times and the use of metaheuristic methods distinguish their work from the current study, which develops an exact method to schedule customer visits by drones and trucks.

Table 4.1. The related studies

Ref.	Problem	Configuration	Time Window	Model	Routing Solution	Scheduling Solution	No. of Customers
(Di Puglia Pugliese & Guerriero, 2017)	FSTSP	1-Truck 1-Drone 1-Depo	Single	MIP	None	None	10
(Ham, 2018)	PDSTSP	m-Truck d-Drone 2-Depo	Single	CP	None	None	100
(Cheng et al., 2020)	DRP	d-Drone 1-Depo	Single	MINLP	B &C	B &C	30
(Di Puglia Pugliese et al., 2021)	VRPD	m-Truck d-Drone 1-Depo	Single	MIP	MSH	MSH	100
(Coindreau et al., 2021)	VRPD	m-Truck d-Drone 1-Depo	Single	MIP	ALNS	ALNS	100
(Kuo et al., 2022)	VRPD	m-Truck d-Drone 1-Depo	Single	MIP	VNS	VNS	100
This study	FSTSP/ PDSTSP	m-Truck d-Drone 1-Depo	Multiple	MIP	ALNS	Exact/CP	100

MIP: Mixed-integer Programming; CP: Constraint Programming; MINLP: Mixed-integer Non-linear Programming; B &C: Branch-and-cut Algorithm; MSH: Multi-start Heuristic; DRP: Drone Routing Problem

The vehicle routing problem with multiple time windows (VRPMTW) has been subject to limited investigation. The VRPMTW arises when the service provider of the last mile delivery operation offers multiple time periods to the customers to deliver their purchases Favaretto et al. (2007) presented a mathematical formulation for VRPMTW, which was solved by ant colony optimization. Belhaiza et al. (2014) proposed a new hybrid variable neighbourhood tabu search heuristic for the VRPMTW. They also presented a minimum backward slack time algorithm to adjust the arrival and departure times to minimize delay during the route generation procedure. Beheshti et al. (2015) proposed a multi-objective mixed-integer programming model for the VRPMTW considering a set of non-overlapping time windows with equal or different lengths. A Cooperative Coevolutionary Multi-objective Quantum-Genetic Algorithm (CCMQGA) is then developed to solve this problem. Hoogeboom et al. (2020) developed an exact polynomial time algorithm to minimize the duration of routes. The current study presents an ALNS algorithm with tailored neighbourhood structure to solve large-size instances. Moreover, an exact approach, Constraint Programming, is applied to find the optimum delivery schedules for the routes generated. Although several studies have been conducted in the VRPTWD literature, there are still gaps that deserve more detailed research.

As can be seen in Table 4.1, none of the previous studies focusing on the VRPTW-D has considered multiple time windows for the customers. Also, none of the studies categorized in VRPTWD literature come up with an exact approach to schedule the visits and synchronize drone-truck interactions.

4.3 MATHEMATICAL PROGRAMMING

This section presents the concepts and equations used in the mathematical formulation. Since our model is based on the concepts presented in the paper by Tamke & Buscher (2021) the notation is similar for ease of comparison. The problem is modelled in the form of an MIP as presented below.

Sets:

C: set of all customers, $C = \{1, \dots, c\}$;

F: set of trucks;

N: set of all nodes, which includes the depot and its dummy representation, $N = \{0\} \cup C \cup \{c+1\}$;

N_0: set of all nodes from which vehicles may depart, $N_0 = N - \{\mathrm{c} + 1\}$;

N^+: set of all nodes except the depot, $N^+ = N - \{0\}$;

D: set of drones;

P: the set P of triples (possible sorties);

T: Set of time windows

Parameters:

τ_{ij}: the travel time from node $i \in N$ to node $j \in N$ for a truck;

τ'_{ij}: the travel time from node $i \in N$ to node $j \in N$ for a drone;

τ_i^{ST}: the service time of trucks at the customer node $i \in C$;

τ_i^{SD}: the service times of drones at the customer node $i \in C$;

τ^L: the required time to launch a drone;

t_{ir}^L: the latest allowed time to visit customer $i \in C$ in time window $r \in T$;

t_{ir}^E: the earliest allowed time to visit customer $i \in C$ in time window $r \in T$;

cv: the truck capacity;

cd: the drone capacity;

w_i: the parcel weight of customer $i \in C$;

rd: the maximum distance that a drone can fly at each operation day;

dis_{ij}: the distance between node $i \in N$ and $j \in N$;

c_{ij}^T: the operating cost of trucks to move from node $i \in N$ to $j \in N$;

c_{ij}^D: the operating cost of drones to move from node $i \in N$ to $j \in N$;

M: a sufficiently large number.

Variables:

x_{ij}^{f}: equals 1 if truck $f \in F$ moves from node $i \in N_0$ to node $j \in N^+$, and 0 otherwise;

y_{ijk}^{fd}: equals 1, if drone $d \in D$ carried by truck $f \in F$ flies from node $i \in N_0$ to visit node $j \in C$, and lands on the truck on node $k \in N^+$, where $(i,j,k) \in P$, and 0 otherwise;

q_{i}^{f}: equals 1 if node $i \in N$ is visited by truck $f \in F$, and 0 otherwise;

b_{ij}^{f}: 1 if nodes $i \in C$ and $j \in C$ ($j > i$) are visited by truck $f \in F$, and 0 otherwise;

u_{i}^{f}: Indicates the position of the customer $i \in C$ on the route of truck $f \in F$, meaning that if $u_i^f < u_j^f$, customer i is visited before customer j, where $u_i^f \in \mathbb{N} \cup \{0\}$);

p_{ij}^{f}: 1 if node $i \in N_0$ precedes node $j \in N^+$ in the tour of truck $f \in F$, and 0 otherwise;

tt_{i}^{f}: the arrival time of truck $f \in F$ at node $i \in N$;

td_{i}^{fd}: the arrival time of drone $d \in D$, dispatched from truck $f \in F$ at node $i \in N$;

t_{i}^{f}: the departure time of truck $f \in F$ from node $i \in N$;

tw_{ir}: equals 1if customer $i \in C$ is visited in time window $r \in T$.

$$\sum_{i \in N_0} \sum_{j \in N^+} \sum_{f \in F} c_{ij}^{T} \times x_{ij}^{f} + \sum_{d \in D} \sum_{i \in N_0} \sum_{j \in C} \sum_{k \in N^+} \sum_{f \in F:(i,j,k) \in P} \left(c_{ij}^{D} + c_{jk}^{D}\right) \times y_{ijk}^{fd} \tag{1}$$

Equation (1) minimises the total logistics cost, comprising two cost components, which correspond to the operation cost of the trucks and drones respectively. The deliveries costs are obtained based on the total distance, where the service costs of drones yield lower operational costs.

Truck routing constraints are:

$$\sum_{j \in N_0} x_{ij}^{f} = q_{i}^{f} \qquad \forall\, i \in N^+, f \in F \tag{3}$$

$$\sum_{j \in N^+} x_{ji}^{f} = q_{i}^{f} \qquad \forall\, i \in N_0, f \in F \tag{4}$$

Equations (3) and (4) represent the truck flow conservation equations.

$$u_{i}^{f} - u_{j}^{f} + |C| \times x_{ij}^{f} + (|C| - 2)x_{ji}^{f} \leq |C| - 1 \qquad \forall\, i,j \in C, f \in F \tag{5}$$

Constraint (5) evaluates the value of u_i^f and u_j^f for costumer i and j on the route of truck $f \in F$, such that if the truck visits customer i before j the u_j^f equals to $u_i^f + 1$.

$$|C| \times p_{ij}^f - (|C| - 1) \leq u_j^f - u_i^f \qquad \forall\, i, j \in C, f \in F \tag{6}$$

Constraint (6) sets the value of precedence variable (p_{ij}^f) based on the values position variables (u_j^f and u_i^f). If the $u_j^f - u_i^f$ is greater than one, costumer j is visited after costumer i according to the route of truck f, forcing variable p_{ij}^f to take value 1.

$$p_{ij}^f + p_{ji}^f = b_{ij}^f \qquad \forall i, j \in C: j > i, f \in F \tag{7}$$

Equation (7) asserts that if customers i and j are served by truck f (b_{ij}^f=1), it requires the truck to visit customer i either before or after costumer j ($p_{ij}^f = 1$ or $p_{ji}^f = 1$).

$$q_i^f \leq b_{ij}^f \qquad \forall i, j \in C: j > i, f \in F \tag{8}$$

$$q_j^f \leq b_{ij}^f \qquad \forall i, j \in C: j > i, f \in F \tag{9}$$

$$q_i^f + q_j^f \leq 1 + b_{ij}^f \qquad \forall i, j \in C: j > i, f \in F \tag{10}$$

Constraints (8)-(10) ensure that costumer i and j are served by truck f(b_{ij}^f=1), provided that those customers are assigned to the truck ($q_i^f = 1$ and $q_j^f = 1$).

$$p_{0,i}^f = q_i^f \qquad \forall\, i \in C, f \in F \tag{11}$$

$$p_{i,|C|+1}^f = q_i^f \qquad \forall\, i \in C, f \in F \tag{12}$$

Equations (11) and (12) control the precedence variable associated with the depot. Equation (11) asserts that the depot must precede all nodes consisting of a truck route. Similarly, Equation (12) ensures that the dummy depot should succeed in all the nodes that appeared in a truck route.

$$\sum_{i \in C} q_i^f w_i + \sum_{i \in N_0} \sum_{j \in C} \sum_{k \in N^+} \sum_{d \in D:(i,j,k) \in P} w_j y_{ijk}^{fd} \leq cv \qquad \forall f \in F \tag{13}$$

Constraint (13) guarantees that the total weight of deliveries assigned to a truck never violates its capacity.

$$\sum_{f\in F} q_j^f + \sum_{i\in N_0}\sum_{k\in N^+}\sum_{d\in D:(i,j,k)\in P} y_{ijk}^{fd} = 1 \qquad \forall\, j\in C \qquad (14)$$

Equation (14) ensures that all the customers are visited either by a drone or by a truck.

Drone routing constraints are:

$$\sum_{j\in C}\sum_{k\in N^+:(i,j,k)\in P} y_{ijk}^{fd} \le q_i^f \qquad \forall i\in N_0, f\in F, d\in D \qquad (15)$$

$$\sum_{i\in N_0}\sum_{j\in C:(i,j,k)\in P} y_{ijk}^{fd} \le q_k^f \qquad \forall k\in N^+, f\in F, d\in D \qquad (16)$$

Constraint (15) and (16) maintain that a truck should visit the retrieving and launching point of drones.

$$\sum_{m\in C}\sum_{n\in N^+:(l,m,n)\in P} y_{lmn}^{fd} \le 3 - \sum_{j\in C:(i.j.k)\in P} y_{ijk}^{fd} - p_{il}^f - p_{lk}^f \qquad \forall l\in C, i\in N_0, k\in N^+, f \in F, d\in D \qquad 17)$$

Constraint (17) prevents drone d to launch again if it is already assigned to a delivery task. Assuming that drone d of truck f is launched from node i to serve customer j and is retrieved at node k ($y_{ijk}^{fd}=1$), this drone cannot start a flight at any intermediate node l that is visited between i ($p_{il}^f=1$) and k (p_{lk}^f=1) on the route of truck f since it is already operating.

$$w_j \times y_{ijk}^{fd} \le cd \qquad \forall i\in N_0, j\in C, k\in N^+, f\in F, d\in D \qquad (18)$$

$$\left(dis_{ij} + dis_{jk}\right)\times y_{ijk}^{fd} \le rd \qquad \forall i\in N_0, j\in C, k\in N^+ \forall f\in F, d\in D \qquad (19)$$

Constraints (18) and (19) guarantee the drone capacity and maximum flight range on each mission are observed, respectively.

Time constraints are:

$$tt_k^f \ge t_i^f + \tau_{ik}\times x_{ik}^f - M\times(1-x_{ik}^f) \qquad \forall\, i\in N_0, k\in N^+, f\in F \qquad (20)$$

According to constraint (20), truck $f\in F$ may arrive node $k\in N^+$ sometime after leaving node $i\in N_0$ at time t_i^f when the travel time between i and k is τ_{ik}.

$$td_j^{fd} \geq tt_i^f + \left(\tau'_{ij} + \tau^L\right) \sum_{k \in N^+:(i,j,k) \in P} y_{ijk}^{fd} - M \times \left(1 - \sum_{k \in N^+:(i,j,k) \in P} y_{ijk}^{fd}\right) \qquad \forall\, i \in N_0, j \in C, f \in F, d \in D \tag{21}$$

$$td_j^{fd} \geq td_i^{fd} + \left(\tau'_{ij} + \tau^L\right) \sum_{k \in N^+:(i,j,k) \in P} y_{ijk}^{fd} - M \times \left(1 - \sum_{k \in N^+:(i,j,k) \in P} y_{ijk}^{fd}\right) \qquad \forall i \in N_0, j \in C, f \in F, d \in D \tag{22}$$

$$td_j^{fd} \leq td_i^{fd} + \left(\tau'_{ij} + \tau^L\right) \sum_{k \in N^+:(i,j,k) \in P} y_{ijk}^{fd} + M \times \left(1 - \sum_{k \in N^+:(i,j,k) \in P} y_{ijk}^{fd}\right) \qquad \forall i \in N_0, j \in C, f \in F, d \in D \tag{23}$$

Constraints (21) and (22) update the drone visiting time based on the earliest departure time at launching node i, which is equal to the maximum of tt_i^f and td_i^{fd} plus the time needed to prepare the launch τ^L. Constraint (23) with constraint (22) ensures that the dispatched drone arrives to the delivery point soon after it is launched.

$$td_k^{fd} \geq td_j^{fd} + \left(\tau_j^{SD} + \tau'_{jk}\right) \sum_{i \in N_0:(i,j,k) \in P} y_{ijk}^{fd} - M\left(1 - \sum_{i \in N_0:(i,j,k) \in P} y_{ijk}^{fd}\right) \qquad \forall j \in C, k \in N^+, f \in F, d \in D \tag{24}$$

$$td_k^{fd} \leq td_j^{fd} + \left(\tau_j^{SD} + \tau'_{jk}\right) \sum_{i \in N_0:(i,j,k) \in P} y_{ijk}^{fd} + M\left(1 - \sum_{i \in N_0:(i,j,k) \in P} y_{ijk}^{fd}\right) \qquad \forall j \in C, k \in N^+, f \in F, d \in D \tag{25}$$

Constraints (24) and (25) determine the collecting time of a dispatched drone, such that the drone flies to the retrieving point without any waiting time or hovering.

$$tt_k^f \leq td_j^{fd} + \left(\tau_j^{SD} + \tau'_{jk}\right) \sum_{i \in N_0:(i,j,k) \in P} y_{ijk}^{fd} + M\left(1 - \sum_{i \in N_0:(i,j,k) \in P} y_{ijk}^{fd}\right) \quad \forall j \in C, k \in N^+, f \in F, d \in D \tag{26}$$

Constraint (26) asserts that the collecting truck should be present at the retrieving point before the drone's arrival.

$$t_i^f \geq tt_i^f + q_i^f \tau_i^{ST} \quad \forall f \in F, i \in N \tag{27}$$

$$t_i^f \geq tt_i^f + \tau^L \sum_{j \in C} \sum_{k \in N^+ : i \in N_0, (i,j,k) \in P} y_{ijk}^{fd} \quad \forall i \in N, f \in F, d \in D \tag{28}$$

$$t_i^f \geq td_i^{fd} + \tau^L \sum_{j \in C} \sum_{k \in N^+ : i \in N_0, (i,j,k) \in P} y_{ijk}^{fd} + \tau_i^{SD} \sum_{h \in N_0} \sum_{k \in N^+ : i \in C, (h,j,k) \in P} y_{hik}^{fd} \quad \forall i \in N, f \in F, d \in D \tag{29}$$

Constraints (27) - (29) determine the time that truck $f \in F$ leaves the node $i \in N$, in tandem with its drones. Constraint (27) defines the leaving time when node i is visited by the truck, and constraint (28) evaluates the departure time when the truck launches a drone in addition to serving costumer i. Constraint (29) specifies the leaving time when a drone is supposed to be retrieved at node $i \in N$.

$$td_j^{fd} \leq \sum_{r \in T} t_{jr}^L \times tw_{jr} + M\left(1 - \sum_{i \in N_0} \sum_{k \in N^+:(i,j,k) \in P} y_{ijk}^{fd}\right) \quad \forall j \in C, d \in D, f \in F \tag{30}$$

$$td_j^{fd} \geq \sum_{r \in T} t_{jr}^E \times tw_{jr} + M\left(1 - \sum_{i \in N_0} \sum_{k \in N^+:(i,j,k) \in P} y_{ijk}^{fd}\right) \quad \forall j \in C, d \in D, f \in F \tag{31}$$

$$tt_j^f \leq \sum_{r \in T} t_{jr}^L \times tw_{jr} + M\left(1 - q_j^f\right) \quad \forall j \in C, f \in F \tag{32}$$

$$tt_j^f \leq \sum_{r \in T} t_{jr}^L \times tw_{jr} + M\left(1 - q_j^f\right) \qquad \forall j \in C, f \in F \tag{33}$$

Constraints (30) and (31) determine the time window at which drones visit customers, and constraints (32) and (33) specify the time window at which trucks visit customers.

$$\sum_{r \in T} tw_{ir} = 1 \qquad \forall i \in C \tag{34}$$

Constraint (34) ensures that only one of the defined customer time windows is picked by the model.

4.4 SOLUTION APPROACH

This section outlines a method for solving the Vehicle Routing Problem with Multiple Delivery Time Windows (VRPDMTW) by utilizing the Adaptive Large Neighbourhood Search (ALNS) algorithm as its core structure. ALNS algorithm, originally introduced by Ropke & Pisinger (2006), selects destroy and repair methods to construct routes. The adaptation procedure is described in Section 4.2. Further information regarding this technique can be found in the works of (Demir et al., 2012; Grangier et al., 2016a; Hemmelmayr et al., 2012; Ribeiro & Laporte, 2012; Ropke & Pisinger, 2006).

The choice of the Adaptive Large Neighbourhood Search (ALNS) algorithm as the core structure for solving the Vehicle Routing Problem with Multiple Delivery Time Windows (VRPDMTW) in this study can be justified based on several compelling reasons:

- Problem Complexity Handling: The VRPDMTW is a highly complex problem, especially when considering the integration of both trucks and drones, multiple time windows, and varying service times. ALNS is well-suited for addressing such intricate problems because it provides a flexible and adaptive framework for exploring various neighbourhood structures. This adaptability allows ALNS to efficiently navigate the problem's complex solution space.
- Effective Route Construction: ALNS combines destroy and repair methods to construct routes. Destroy methods help break down and restructure sub-optimal routes, while repair methods work to improve them. This combination of strategies enables ALNS to iteratively refine and optimize routes, leading to high-quality solutions.

- Previous Successes: ALNS has demonstrated its effectiveness in solving various vehicle routing problems in previous research, as referenced in works by Chen et al. (2021), Sacramento et al. (2019), and Jun Li et al. (2022). The algorithm has a track record of producing competitive solutions, making it a reliable choice for tackling the VRPDMTW.
- Adaptation: As the name suggests, ALNS is adaptive and can be tailored to the specific characteristics of the problem at hand. This adaptability is crucial when dealing with real-world logistics problems, as it allows for customization to match the problem's unique constraints and requirements.

The proposed approach comprises of two main phases to solve the problem. In the first phase, the best truck route to visit all customers is determined and in the second phase an attempt is made to assign as many customers as possible to drones, with the aim of minimizing the total cost. In each phase, several simple yet effective heuristics are applied as repair and destroy functions to improve the initial solution. Figure 4.2 displays the overall steps of the proposed solution approach.

The procedure for constructing the initial solution is explained in Section 4.4.1, and the parameter adaptation procedure is described in Section 4.4.2. Then, the destroy and repair operators for phases 1 and 2 are presented in Section 4.4.3 and 4.4.4, respectively. Section 4.4.5 presents an exact approach formulated as a Constraint Programming model to find the optimum scheduling decisions for a given route.

4.4.1 Initial solution

The initial truck route is constructed using a sequential insertion technique that builds one route at a time until all customers are visited. The construction process starts with a round trip from the depot to a seed node, chosen as the unassigned customer that minimizes $100 \times (t_{i1}^{L} - t_{i1}^{E}) + \tau_{i0}$. Customers are added to the route in the order of least tour cost increase until a constraint is violated, at which point a new route is started with a new seed node. The initial route is then refined using two successive improvement processes that involve relocating a random customer within a route in Phase 1 and swapping two randomly chosen customers in Phase 2. Both phases are repeated for a fixed number of outer iterations, with feasibility conditions being checked in each attempt.

4.4.2 Adaptation procedure

After a fixed number of iterations of the destroy/repair cycle, the adaptation procedure is applied in an outer iteration. The probability of selecting a given destroy or repair operator is updated based on its score, which measures its performance in improving solutions. The scores are initially set to zero, but if a destroy/repair operator results in a new overall best solution, its score is incremented by a parameter χ_1. If the iteration results in a new incumbent solution, the operators receive a reward score of χ_2, and if the new solution has not previously been seen, the destroy and repair operators are rewarded with a score of χ_3. In each iteration, the probability of selecting each operator is updated to be directly proportional to its score, while maintaining a lower bound on the probability. The algorithm biases the selection of operators based on their performance in different customer distributions, making it robust to various situations.

4.4.3 Operators used in generating a new solution at Phase 1

The destroy and repair operators used in this phase are the slightly modified version of those introduced by Ropke & Pisinger (2006). At the end of this section, the mechanism by which each operator is applied will be explained. We encourage the interested reader to see (Ropke & Pisinger, 2006) for a detailed explanation.

4.4.3.1. Destroy Methods

The first phase considers four destroy methods that remove customers from truck routes and assign them to drones. These methods take the number of nodes to remove and the truck routes as input and output a partial solution with q customers removed from the complete solution. The removed customers are stored in a list called the *removal list*.

Shaw Removal Heuristic

The main idea of this operator is the same as developed by Ropke & Pisinger (2006). This operator selects a random customer i, then uses equation (35) to calculate its relatedness to other customers. Then, q customers are randomly unassigned and added to the *removal list* based on their relatedness, where those with a high relatedness value are more likely to be removed. The relatedness between customers is determined by considering the cost of travel between them, the difference in visit times, the difference in time windows, and the number of trucks that can serve them.

Regenerate response:

$$R(i,j) = \varphi_1(d_{ij}) + \varphi_2(|T_i - T_j|) + \varphi_3(|l_i - l_j|) + \varphi_4\left(1 - \frac{|K_i \cap K_j|}{min\{|K_i|, |K_j|\}}\right) \quad (35)$$

where T_i represents the visiting time of customer i, K_i indicates the set of trucks (truck routes) which are able to serve customer i. Note that φ_1 to φ_4 are the coefficients of each of the introduced four criteria, respectively**.**

Random Removal

This operator randomly removes q customers from the truck routes to explore the search space and avoid local optimal solutions. While it has a low chance of improving the solution, its random search scheme can be effective to evade from early stagnation.

Worst Cost Removal

This operator selects the customers whose removal will result in the greatest cost reduction (saving). In other words, all visited customers in the current solution s are checked and for each customer i, $\Delta f_i^{-f(s)-f(s/\{i\})}$ is calculated. Then the values of Δf^- are sorted and the q customer are randomly selected to be removed from the solution s based on their removal chance. The customers with larger values for Δf^- have a higher probability of being removed.

Route Removal

This operator randomly removes q customers from a randomly selected route. To improve its performance, it considers different probabilities for destroying each route based on the number of customers it serves. The fewer customers a route has, the higher its probability of being destroyed. The Roulette Wheel mechanism is used to select the route and randomly remove q customers from it. If q is greater than the number of customers in the selected route, another route is chosen using the same mechanism until the number of deleted customers equals q.

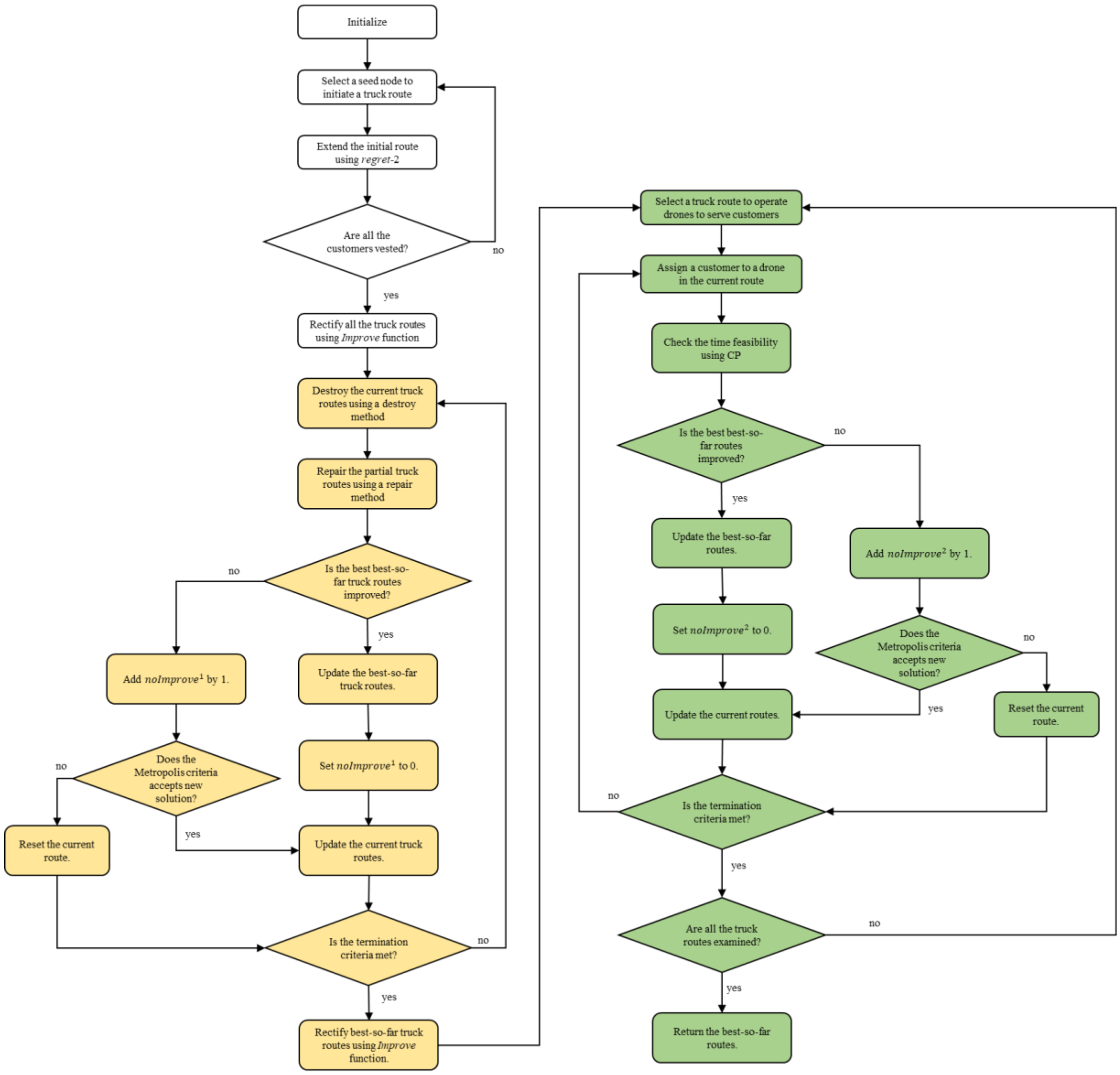


Figure 4.2. A flowchart of the proposed approach. Phase 1 is shown by the yellow backgrounds, and Phase 2 is indicated by the green backgrounds. The white objects illustrate the procedures for creating the initial solution.

4.4.3.2. Repair methods

After the destroy operators, repair operators are applied to insert customers back into the solution. Feasible insertions must adhere to capacity limits and time window considerations. Two types of repair operators, Greedy Insertion and Regret-k Insertion are used in phase 1.

Greedy Insertion

This operator searches all available positions for each customer in the *removed list* and performs the insertion that creates the minimum incremental cost. After reinserting all new customers existing in the removal list, a new complete solution is obtained. This operator only considers the trucks, while the drones are treated in the later phase.

Regret-k Insertion

Regret-*k* insertion is an operator that uses look-ahead information to select a customer to insert. Regret-2 calculates the amount of regret for each customer by identifying the location where inserting the customer leads to the least cost increase in each truck route and then defining the regret as the difference in cost of inserting the customer in the first best route and the second-best route. The customer with the highest regret is inserted in the first best route until there are no more nodes to insert or insertion is no longer possible for existing routes. Regret-*k* can be generalized by calculating the difference in cost of inserting customer i in the first best route with the k^{th} best route. In this study, the values of 2 and 3 are selected for k.

4.4.4 Operators used in generating a new solution at phase 2

After constructing the truck routes in the first phase, in the second phase, the algorithm seeks to find an allocation of customers to drones.

4.4.4.1. Destroy methods

In the second phase, only one destroy function is considered to remove customers from the truck routes, which takes the number of nodes to be removed and a truck route as inputs.

Leave truck route

This function takes truck routes as input, removes a random customer, checks for drone capacity, and adds it to the removal list. If the customer is a drone dispatch or collection point, the drone's route is also deleted, and its assigned customer is added to the list.

Drone assignment

This operator assigns unassigned customers to drones starting from the first customer in the removal list. If any customers remain unassigned, the solution is considered infeasible. Once the routes are generated, a Constraint Programming model (CP) is used to schedule the drones and trucks.

4.4.5 Constraint Programming model

This section presents a Constraint Programming (CP) model for the truck-drone scheduling problem with multiple time windows, aiming to minimize the total delivery time. Unlike heuristic approaches, which cannot prove optimality, this study proposes an exact approach based on mathematics (Baptiste et al., 2001). CP is a powerful tool that has been successfully applied to many optimization problems, particularly in scheduling (Rossi et al., 2006). The CP model takes a sequence of visits as input, with each visit specifying the customer and the visiting vehicle (i.e. drone or truck). It checks feasibility and determines the timing of the given sequence, producing the time of each visit, the arrival and departure time of the truck, and the launching and collecting time of drones. The notation used in the CP model is provided below.

Set:

V: sequence of the visits, where the first visit and the last visits are the depot, $V \in \{1,2,\dots,|V|\}$;

A_i: Represents the customer of visit i.

parameters:

λ_i: determines the visitor of visit $i \in V$, $\delta_i \in \{Drone, Truck\}$;

γ_i^T: represents the travel time of a truck to get to visit $i \in V$ from its previous visit;

γ_i^D: represents the travel time of a drone to get to visit $i \in V$ from its previous visit;

π_i: represents the previous visit to $i \in V$;

η_i: represents the next visit to visit $i \in V$;

Variables:

α_i: represents the arrival time of visit $i \in V$;

σ_i: represents the starting time of serving the visit $i \in V$;

δ_i: represents the departure time from visit $i \in V$;

Equation (36) is the objective of the CP model, which is to minimizes the time required to serve all visits in V:

$$\min \alpha_{|V|} - \delta_1 \tag{36}$$

Constraint (37) ensures that visit $i \in V$ starts within one of the defined time windows:

$$\text{For any } r \in T \text{ holds } \left(t_{(A_i)r}^E \le \sigma_i \le t_{(A_i)r}^L\right) \qquad \forall i \in V \tag{37}$$

Constraint (38) implies visit i starts as soon as the drone arrives, but the truck can wait for a while until the time window starts.

$$\text{if } \lambda_i = Drone \text{ then } \alpha_i = \sigma_i, \text{ else } \alpha_i \leq \sigma_i \qquad \forall i \in V \qquad (38)$$

Constraint (39) guarantees that drones are launched when the truck is still located at the previous visit.

$$\text{if } \lambda_i = Drone \text{ then } (\alpha_{\pi_i} \leq \alpha_i - \gamma_i^T \text{ and } \alpha_i - \gamma_i^T \leq \delta_{\pi_i}) \qquad \forall i \in V \qquad (39)$$

Therefore, a drone dispatching time should be after the truck's arrival time and before the departure time of the last visit. For security, drones should fly to the retrieving point when a truck is available at that point, which is guaranteed by Constraint (40):

$$\text{if } \lambda_i = Drone \text{ then } (\alpha_{\eta_i} \leq \delta_i + \gamma_i^D \text{and } \delta_i + \gamma_i^D \leq \delta_{\eta_i}) \qquad \forall i \in V \qquad (40)$$

Constraint (41) implies that drones should leave a visit immediately after their service is finished. The truck, on the other hand, may stay longer after its serving time is over:

$$\text{if } \lambda_i = Drone \text{ then } \sigma_i + \tau_{(A_i)}^{SD} = \delta_i, \text{ else } \sigma_i + \tau_{(A_i)}^{ST} \leq \delta_i \qquad \forall i \in V \qquad (41)$$

Equation (42) calculates the arrival time of the truck at each visit:

$$\text{if } \lambda_i = Truck \text{ then } \alpha_i = \delta_{\pi_i} + \gamma_i^T \qquad \forall i \in \{2,3, \dots, |V| - 1\} \qquad (42)$$

4.5 RESULTS AND ANALYSES

The VRPDMTW problem is complex and global optimality solvers cannot handle large networks, which are often encountered in real-world applications. To ensure our algorithm can produce high-quality solutions within reasonable timeframes for large networks, we created a new dataset based on Solomon's VRPTW dataset (Solomon, 1987). As the first study to incorporate multiple time windows in the VRPD considering all the assumptions made, no relevant benchmark dataset exists.

The study Implemented all procedures in Java on an x64-PC with an Intel Core i7-7600 2.3 GHz CPU and 8.0 GB of RAM, using Gurobi version 9.0.2 as the exact solver. Test instance formation is described in Section 4.5.1, algorithm parameter settings are presented in Section 4.5.2, and the experimental findings are discussed in Section 4.5.3. Management perspectives are provided in Section 4.5.4.

4.5.1 Test instances

Solomon (1987) test instances represent various aspects of VRPTW, including geographic data, customer quantity, and time window attributes. Customers are located within a $[0,100]^2$ square and are divided into six test sets, R1, R2, C1, C2, RC1, and RC2. R1 and R2 randomly generate geographic information, while customers are clustered in C1 and C2. In instances of type 1 (R1, C1, RC1), all customers have narrow time windows, while instances of type 2 have a longer horizon.

To examine the implications of multiple time windows, we assigned up to three non-overlapping time windows to each customer, taking Solomon's time windows as a starting point. Algorithm 4.1 outlines the procedures used to generate multiple time windows.

Two sets of VRPDMTW instances are generated using Solomon (1987) benchmark and Algorithm 4.1, comprising 12 and 100 customers for small and large instances, respectively. Solomon (1987) benchmark instances for the VRPTW are used to evaluate the ALNS algorithm's performance.

Algorithm 4.1. Time window generation.

GenerateTimes ($E =${The starting time of Solomon’s time windows}, $L =$ {The Latest time of Solomon’s time windows}, $|R| \in N$):
  $r \leftarrow 1$;
  the new time window $tw_{ir} \leftarrow \{E_i, L_i\} \forall i \in N_0$;
  for each ($i \in N_0$) do:
    $k \leftarrow 2$;
    while ($k \leq 2|R|$) do:
      $r \leftarrow \lfloor \frac{k}{2} \rfloor + 1$
      a ←the last element in $tw_{i(r-1)}$;
      A ←the length of the last time window in $tw_{i(r-1)}$
      η ←a random number in $[a + \frac{A}{2}, a + 2A]$;
      $tw_{ir} \leftarrow tw_{ir} \cup \eta$;
      $k \leftarrow k + 1$;
    end while;
  end for;
return tw_{ir}.

The Euclidean distance metrics are used to calculate distances for drones and trucks. Since the drone-truck delivery system is not yet widely applied and is in the early stages of development, accurate extraction of the required parameters is not possible. Therefore, all parameters have been extracted from the relevant literature and their accuracy estimated with care and tested for sensitivity.

It is assumed that the drones have a flight range of 2.5 km and are capable of carrying packages with a maximum weight of 10 kg. Each truck has a maximum payload of 700 kg, each of which is equipped with two drones. The fuel consumption is 0.15 litter/km (≈0.24 liter/mile) for a truck and oil cost is $0.83 per liter (*Gasoline Prices around the World, 02-Nov-2020 | GlobalPetrolPrices.Com*, n.d.). Drone operating cost is set to $0.05 per km (Kim 2016, n.d.). The speeds of truck and drones are assumed to be 35 mph (≈ 56 kph) and 50 mph (≈ 80 kph), respectively (Sacramento et al., 2019b). The service time for visiting a customer is 5 minutes.

Table 4.2. Parameter setting in the proposed ALNS algorithm

Parameter	Description	Value
n	Number of iterations in the improvement function	100
n^1	Number of iterations for relocating a random customer in the improvement function	100
n^2	Number of iterations for swapping two random customers in the improvement function	100
φ_1	Coefficient of spatial distance in Shaw removal function	9
φ_2	Coefficient of visiting time difference in Shaw removal function	3
φ_3	Coefficient of time window distance in Shaw removal function	2
φ_4	Coefficient of available trucks in Shaw removal function	5
χ_1	Reward for the heuristic which improved the best solution	33
χ_2	Reward for the heuristic which find a new solution accepted by metropolis criteria	9
χ_3	Reward for the heuristic which find a new solution	13
q	The number of removed customers at each iteration	$U\left[\frac{\lvert N\rvert}{20},\frac{\lvert N\rvert}{3}\right]$
k	The value for regret-*k* method	{1,2,3}
$maxNoImprove^1$	The termination criteria for the phase 1	100
$maxNoImprove^2$	The termination criteria for the phase 2	10
p	The probability threshold for choosing a candidate node	0.20

The method outlined by Ropke and Pisinger (2006) is applied to tune the ALNS's settings, using a subset of instances from each set (C1, C2, R1, R2, RC1, and RC2). The best parameter value is selected from three executions for each parameter, and the parameter settings for the modified instances are listed in Table 4.2. Most parameter values used are from (Sacramento et al., 2019) study.

4.5.2 Performance comparison of heuristics for small network instances

This section evaluates the ALNS algorithm's performance on small VRPDMTW networks by generating 58 instances with 12 customers and a depot. The networks are divided into three configurations (MC, MR, and MRC) based on customer clustering and distribution, following Solomon (1987) instance generation process using Algorithm 4.1. The solving algorithm is compared to the MIP model. Each grid is run 10 times by ALNS algorithm and the results consisting of best solution (Z^{ALNS}), average solution (μ^{ALNS}), averaged execution time (t^{ALNS}), and the standard variation (σ^{ALNS}) are shown in Table 4.3. The performance of the algorithm is compared with the optimal solution (Z^{MIP} obtained by solving the MIP model according to the two measures of execution time) and optimality gap (ρ_{ALNW}^{MIP}), as measured by the ratio of the metaheuristic's average value to the optimal solution is determined by equation (43):

$$\rho_{ALNW}^{MIP} = \left(1 - \frac{\mu^{ALNS}}{Z^{MIP}}\right) \times 100\% \tag{43}$$

Table 4.3 shows that the proposed ALNS algorithm is capable of converging to the optimal solution in most instances. Furthermore, the average objective value (μ^{ALNS}) is equal to the best solution (Z^{ALNS}) in most cases, which indicates the robustness of the algorithm being able to provide good solutions consistently. The average of optimality gap for all instances is 0.15% with an average of roughly 10.75 seconds execution time.

Table 4.3. Performance of the ALNS algorithm for small instances. The column t^{MIP} represents the CPU time (in seconds) in GORUBI for obtaining the optimal solution, whereas the column t^{ALNS} shows the average amount of CPU time (in seconds) employed by ALNS algorithm to obtain the best-known solutions. The column $C\vee$ reports the number of customers.

Scenario		MIP		ALNS				
Config.	$C\vee$	Z^{MIP}	t^{MIP}	Z^{ALNS}	t^{ALNS}	μ^{ALNS}	σ^{ALNS}	$\%\rho_{ALNW}^{MIP}$
MC101	10	94.09	386.68	94.09	9.90	94.09	0.00	0.00
MC102	10	91.85	609.13	91.85	7.75	91.85	0.00	0.00
MC103	10	91.85	470.60	91.85	8.61	91.85	0.00	0.00
MC104	10	91.85	562.05	91.85	9.39	91.85	0.00	0.00
MC105	10	92.76	651.94	92.76	12.61	92.76	0.00	0.00
MC106	12	93.06	464.28	93.06	7.46	93.06	0.00	0.00
MC107	12	92.76	595.40	92.76	11.57	92.76	0.00	0.00
MC108	12	91.85	624.70	91.85	15.88	96.85	3.49	0.00
MC109	15	92.81	384.64	95.81	13.26	99.81	5.96	3.23

MC201	10	167.60	595.48	167.6	12.50	167.6	0.00	0.00
MC202	10	164.54	717.40	164.54	12.18	164.54	0.00	0.00
MC203	10	165.02	574.25	165.02	7.95	165.02	0.00	0.00
MC204	10	160.40	672.59	160.4	7.23	160.4	0.00	0.00
MC205	10	164.54	665.34	164.54	8.30	164.54	0.00	0.00
MC206	12	158.76	744.52	158.76	11.38	162.76	6.12	0.00
MC207	12	161.82	415.14	163.76	9.21	163.76	0.00	1.20
MC208	15	159.64	380.21	159.64	14.42	159.64	0.00	0.00
MR101	10	211.09	1759.28	211.09	12.61	211.09	0.00	0.00
MR102	10	225.63	1610.59	225.63	12.69	225.63	0.00	0.00
MR103	10	203.95	1228.52	203.95	7.29	203.95	0.00	0.00
MR104	10	178.61	924.62	178.61	11.62	178.61	0.00	0.00
MR105	10	161.42	1224.49	161.42	10.17	161.42	0.00	0.00
MR106	10	161.57	981.65	161.57	9.40	161.57	0.00	0.00
MR107	10	214.33	1141.34	214.33	8.84	214.33	0.00	0.00
MR108	12	186.22	1698.01	186.22	11.43	186.22	0.00	0.00
MR109	12	211.09	1080.79	211.09	10.23	214.09	5.26	0.00
MR110	12	177.13	1334.06	177.13	8.37	177.13	0.00	0.00
MR111	15	205.30	1647.19	208.74	14.38	213.74	6.67	1.68
MR112	15	161.57	1591.37	161.57	15.28	163.57	3.33	0.00
MR201	10	298.07	955.88	298.07	8.42	298.07	0.00	0.00
MR202	10	220.01	932.21	220.01	12.84	220.01	0.00	0.00
MR203	10	214.87	1028.87	214.87	12.81	214.87	0.00	0.00
MR204	10	184.13	1575.20	184.13	10.30	184.13	0.00	0.00
MR205	10	257.70	1718.12	257.7	11.46	257.7	0.00	0.00
MR206	10	188.16	928.42	188.16	11.61	188.16	0.00	0.00
MR207	10	182.18	982.93	182.18	11.48	182.18	0.00	0.00
MR208	12	144.30	1369.75	144.3	9.36	144.3	0.00	0.00
MR209	12	187.37	1537.96	187.37	9.35	187.37	0.00	0.00
MR210	12	204.29	1273.01	204.29	8.66	204.29	0.00	0.00
MR211	10	182.99	1078.55	182.99	10.60	182.99	0.00	0.00
MRC101	10	193.64	262.26	193.64	9.06	193.64	0.00	0.00
MRC102	10	178.38	272.97	178.38	11.05	178.38	0.00	0.00
MRC103	12	178.38	213.44	178.38	7.75	178.38	0.00	0.00
MRC104	12	172.89	259.00	172.89	12.43	172.89	0.00	0.00
MRC105	12	184.03	168.14	184.03	10.80	184.03	0.00	0.00
MRC106	10	185.85	229.48	185.85	12.69	185.85	0.00	0.00
MRC107	10	175.93	258.46	175.93	11.10	175.93	0.00	0.00
MRC108	10	172.89	221.53	172.89	10.93	172.89	0.00	0.00
MRC201	10	267.04	250.57	267.04	12.88	267.04	0.00	0.00

MRC202	10	210.75	177.56	210.75	7.55	210.75	0.00	0.00
MRC203	10	210.75	274.54	210.75	10.26	210.75	0.00	0.00
MRC204	10	144.16	267.50	144.16	10.39	144.16	0.00	0.00
MRC205	12	177.40	195.70	177.4	10.54	177.4	0.00	0.00
MRC206	12	188.54	247.50	193.12	9.82	193.12	0.00	2.43
MRC207	12	181.03	261.49	181.03	11.58	181.03	0.00	0.00
MRC208	15	142.49	256.36	142.49	14.68	142.49	0.00	0.00
Mean		172.52	766.67	172.75	10.75	173.16	0.55	0.155

The results indicate that the execution time of the MIP model is quite high, even for these relatively small instances. The ALNS algorithm, however, is far less dependent on the problem size: None of the networks required more than 16s to solve.

4.5.3 Performance comparison of ALNSCW algorithm for large network instances

In this section, the performance of the ALNS algorithm is examined for large instances. As no prior study has considered multiple time windows in the VRPD, there is no existing benchmark dataset to compare our results against. For this, 58 benchmark instances, provided by Solomon (1987), containing 100 customers each dispersed over $[0,100]^2$ square, are considered. Note that the purpose of comparing this algorithm with the Solomon (1987) benchmark is purely to test the performance of the algorithm, and the development of an optimization algorithm for VRPTW for beating the best heuristics is not in the scope of this study.

Since the MIP model cannot be used to measure the metaheuristic's effectiveness for larger instances due to the computational complexity, the solutions are compared with those offered by Solomon (1987) for the VRPTW.

The ALNSCW algorithm is applied 10 times to each instance. Since the instances provided by Solomon (1987) are for the truck-only routing problem with time window (VRPTW), we exclude the drone operations in order to test the performance of the algorithm with a set of comparable benchmarks. The results of the performance comparison of the proposed algorithm and the benchmarks are reported in Table 4.4. In addition to Solomon's results, the proposed algorithm is compared with the best-known solutions identified by heuristics, and the optimum solutions. Note that, for some instances, the optimum solutions are not available. Afterwards, the gaps between the best (Z^{ALNS}) objective values of ALNS to the optimal solution (Z^{OPT}), results obtained by best-known heuristics (Z^{BH}) and Solomons' solutions (Z^{SOL}) are calculated by equations (43)-(45).

$$\rho_{ALNW}^{Sol} = 1 - \frac{Z^{ALNS}}{Z^{SOL}} \tag{44}$$

$$\rho_{ALNW}^{BH} = 1 - \frac{Z^{ALNS}}{Z^{BH}} \tag{45}$$

$$\rho_{ALNW}^{OPT} = 1 - \frac{Z^{ALNS}}{Z^{OPT}} \tag{46}$$

The main objective of this study is to achieve quality solutions for the VRPDMTW in a reasonable CPU time. However, as seen in Table 4.4, in some instances, the algorithm is able to provide better solutions with the same number of vehicles used compared to Solomon's benchmarks. For instance, in networks R201 and RC201, the proposed ALNS algorithm outperforms the best-known heuristics provided with the same number of vehicles employed.

Table 4.4. Performance comparison for large instances. The column Z^{ALNS} represents the best objective obtained the ALNS, whereas the column Z^{SOL} , Z^{BH}, and Z^{OPT} show the objective function values obtained by the Solomon, Best Known Solutions Identified by Heuristics, and optimum solutions, respectively. The columns ρ_{ALNW}^{Sol} and ρ_{ALNW}^{BH} , and ρ_{ALNW}^{OPT}report the gap identified by comparing the ALNS solutions with Solomon, Best heuristics (BH), and the optimum solutions, respectively.

	ALNS			Solomon		BH	Optimal			
Ins.	N^T	Z^{ALNS}	σ^{ALNS}	N^T	Z^{SOL}	Z^{BH}	Z^{OPT}	ρ_{ALNW}^{Sol}	ρ_{ALNW}^{BH}	ρ_{ALNW}^{OPT}
C101	10	828.94	4.00	10	828.94	828.94	827.30	0.00	0.00	0.00
C102	10	828.94	14.51	10	828.94	828.94	827.30	0.00	0.00	0.00
C103	10	828.06	15.56	10	828.06	828.06	826.30	0.00	0.00	0.00
C104	10	824.78	21.76	10	824.78	824.78	822.90	0.00	0.00	0.00
C105	10	828.94	16.51	10	828.94	828.94	827.30	0.00	0.00	0.00
C106	10	828.94	13.93	10	828.94	828.94	827.30	0.00	0.00	0.00
C107	10	828.94	14.33	10	828.94	828.94	827.30	0.00	0.00	0.00
C108	10	828.94	13.06	10	828.94	828.94	827.30	0.00	0.00	0.00
C109	10	828.94	14.22	10	828.94	828.94	827.30	0.00	0.00	0.00
C201	3	591.56	13.37	3	591.56	591.56	589.10	0.00	0.00	0.00
C202	3	591.56	16.71	3	591.56	591.56	589.10	0.00	0.00	0.00
C203	3	591.17	16.13	3	591.17	591.17	588.70	0.01	0.01	0.01
C204	3	593.93	19.27	3	590.6	590.6	588.10	0.00	0.00	0.00
C205	3	588.88	13.69	3	588.88	588.88	586.40	0.00	0.00	0.00
C206	3	588.49	11.12	3	588.49	588.49	586.40	0.00	0.00	0.00
C207	3	588.29	12.10	3	588.29	588.29	585.80	0.00	0.00	0.00
C208	3	588.32	11.83	3	588.32	588.32	585.80	0.02	0.02	0.03
R101	20	1679.04	19.07	19	1650.8	1645.79	1637.70	0.04	0.04	0.05

R102	20	1541.49	13.37	17	1486.12	1486.12	1466.60	0.00	0.00	0.07
R103	13	1294.94	16.13	13	1292.68	1292.68	1208.70	-0.03	-0.03	NA
R104	10	977.53	21.44	9	1007.31	1007.24	971.5	0.01	0.01	0.03
R105	14	1393.55	19.85	14	1377.11	1377.11	1355.30	0.01	0.01	0.02
R106	14	1260.19	20.56	12	1252.03	1251.98	1234.60	-0.01	-0.01	0.03
R107	11	1091.56	18.45	10	1104.66	1104.66	1064.60	-0.01	-0.01	NA
R108	10	951.99	21.34	9	960.88	960.88	NA	-0.01	-0.01	0.03
R109	12	1179.09	22.11	11	1194.73	1194.73	1146.90	-0.01	-0.01	0.04
R110	11	1107.72	19.30	10	1118.84	1118.59	1068.00	-0.04	-0.04	0.00
R111	11	1048.04	21.58	10	1096.72	1096.72	1048.70	-0.02	-0.02	NA
R112	10	965.83	23.15	9	982.14	982.14	NA	-0.02	-0.02	0.07
R201	4	1222.41	18.61	4	1252.37	1252.37	1143.20	-0.07	-0.07	NA
R202	4	1104.09	21.28	3	1191.7	1191.7	NA	-0.03	-0.03	NA
R203	4	913.81	19.05	3	939.5	939.54	NA	-0.08	-0.08	NA
R204	3	757.13	21.06	2	825.52	825.52	NA	-0.01	-0.01	NA
R205	4	984.34	24.30	3	994.43	994.42	NA	0.01	0.01	NA
R206	3	915.37	21.23	3	906.14	906.14	NA	-0.08	-0.08	NA
R207	3	818.82	18.85	2	890.61	893.33	NA	-0.02	-0.02	NA
R208	2	713.72	22.79	2	726.82	726.75	NA	0.03	0.03	NA
R209	3	936.99	21.03	3	909.16	909.16	NA	-0.01	-0.01	NA
R210	4	929.12	21.40	3	939.37	939.34	NA	-0.10	-0.11	NA
R211	3	795.50	21.47	2	885.71	892.71	NA	-0.02	-0.02	0.03
RC101	16	1662.73	17.77	14	1696.95	1696.94	1619.80	-0.04	-0.04	0.03
RC102	13	1495.32	23.56	12	1554.75	1554.75	1457.40	0.01	0.01	0.01
RC103	12	1270.81	24.99	11	1261.67	1261.67	1258.00	0.00	0.00	NA
RC104	10	1137.03	21.98	10	1135.48	1135.48	NA	-0.05	-0.05	0.02
RC105	14	1550.57	16.48	13	1629.44	1629.44	1513.70	-0.03	-0.03	NA
RC106	13	1383.65	23.64	11	1424.73	1424.73	NA	0.00	0.00	0.02
RC107	12	1235.16	23.18	11	1230.48	1230.48	1207.80	0.02	0.02	0.05
RC108	11	1167.72	21.59	10	1139.82	1139.82	1114.20	-0.05	-0.05	0.06
RC201	4	1340.17	23.69	4	1406.94	1406.91	1261.80	-0.15	-0.15	0.06
RC202	4	1161.29	24.71	3	1365.65	1367.09	1092.30	-0.09	-0.09	NA
RC203	4	956.81	20.27	3	1049.62	1049.62	NA	0.01	0.01	NA
RC204	3	804.26	19.19	3	798.46	798.41	NA	-0.03	-0.02	0.10
RC205	5	1265.18	21.92	4	1297.65	1297.19	1154.00	-0.04	-0.04	NA
RC206	4	1105.50	29.02	3	1146.32	1146.32	NA	-0.04	-0.04	NA
RC207	4	1019.37	26.46	3	1061.14	1061.14	NA	0.02	0.02	NA
RC208	3	841.24	24.57	3	828.14	828.14	NA	-0.02	-0.02	NA

4.5.4 Managerial discussions

This section provides managerial insights drawn from the computational experiments. While acknowledging Sacramento et al. (2019) discussion on various VRPD parameters related to drones, this study's main contribution is a novel approach to solving the multiple time windows problem. Therefore, we particularly focus on analysing the impact of time windows on drone-truck delivery operations.

4.5.4.1. Impact of the presence of multiple time windows in the VRPDTW

In this section, the performance of the algorithm when considering multiple time windows is examined for different problem settings. Hence, the proposed algorithm is applied for the VRPDMTW, the VRPDTW, and the VRPTW, and the savings obtained with respect to the objective value are reported accordingly. To solve the VRPDTW and the VRPTW, we use the geographic data and the time windows published by Solomon (1987). To generate multiple non-overlapping time windows for solving VRPDMTW, the same process described in Algorithm 4.1 is used. Note that the VRPDMTW, the VRPDTW, and the VRPTW all share exactly one time window. To investigate the impact of the presence of multiple time windows, 58 sets of instances according to the prescribed approach are generated. Each of the network instances is run 10 times and the best objective found is then stored as $Z_{VRPDMTW}$, Z_{VRPDTW}, and Z_{VRPTW} for the VRPDMTW, the VRPDTW, and the VRPTW, respectively. The difference between the VRPDMTW and the VRPDTW is in the number of time windows associated with each customer. To calculate Z_{VRPTW}, the drones are excluded and only trucks are taken into account.

The average savings obtained by comparing different routing platforms are reported in Table 4.5 (%Saving), and the potential savings for different network configurations are shown in Figure 4.3. The results indicate that incorporating multiple time windows and drones in the VRPTW can save costs by around 29% on average, while the VRPDMTW with multiple time windows can achieve a savings of around 16% compared to the VRPDTW. The deployment of drones results in a total cost savings of around 12% for VRPTW with one time window, compared to 9% for the VRPDMTW with multiple time windows.

Table 4.5. The percentage of the potential savings obtained by comparing different routing platforms

	%Saving			
Networks	VRPDMTW/VRPDTW	VRPDMTW/VRPMTW	VRPDTW/VRPTW	VRPDMTW/VRPTW
C1	0.9%	2.9%	2.5%	3.4%

C2	0.8%	6.1%	5.7%	6.6%
R1	24.1%	10.0%	16.0%	41.7%
R2	19.9%	18.7%	20.3%	39.9%
RC1	23.7%	5.3%	13.6%	38.9%
RC2	29.4%	12.7%	13.0%	42.8%
Mean	16.5%	9.3%	11.8%	28.9%

In Figure 3(a), the benefits of using hybrid drone-truck delivery operations are shown for all network configurations. Figure 3(b) demonstrates the impact of utilizing drones in networks with multiple time windows. The width of time windows affects the synchronization requirements between trucks and drones, resulting in limited drone contribution for datasets with short time windows (R1, C1, and RC1). Figure 3(c) displays the effects of drones with multiple time windows on delivery missions for various networks. Finally, Figure 3(d) indicates that incorporating multiple time windows in drone operations further reduces the overall delivery cost.

Comparing Figures 3(b) and 3(d), reveals that multiple time windows can have a greater impact on cost reduction than the use of drones in delivery operations. Overall, using drones results in cost savings for all network configurations, with greater savings for randomly dispersed nodes. Savings are lower for constrained time windows in C1 and RC1 datasets, but using drones and multiple time windows can still result in a significant decrease in operational costs.

Savings of up to 63% can be achieved by comparing the VRPDMTW with the VRPTW for randomly distributed networks. In networks with random distribution, the widest variety of variances may be seen (R1,R2). Additionally, these networks achieve the highest savings amounts. The clustered network distribution (C1,C2), in comparison, shows the least cost savings and variability. This can be rationalized by the fact that in clustered networks, the placement of one- or multiple-time windows does not significantly affect the order of customer visits, since customers belonging to a certain cluster are located relatively close to each other. Assuming that the geographical locations of demand points in networks R1, R2, RC1 and RC2 are generated in such a way that the distribution of customers within the network is random or semi-random, greater flexibility in the customers' time windows enables the algorithms to change the routes of the vehicles, leading to significant cost savings.

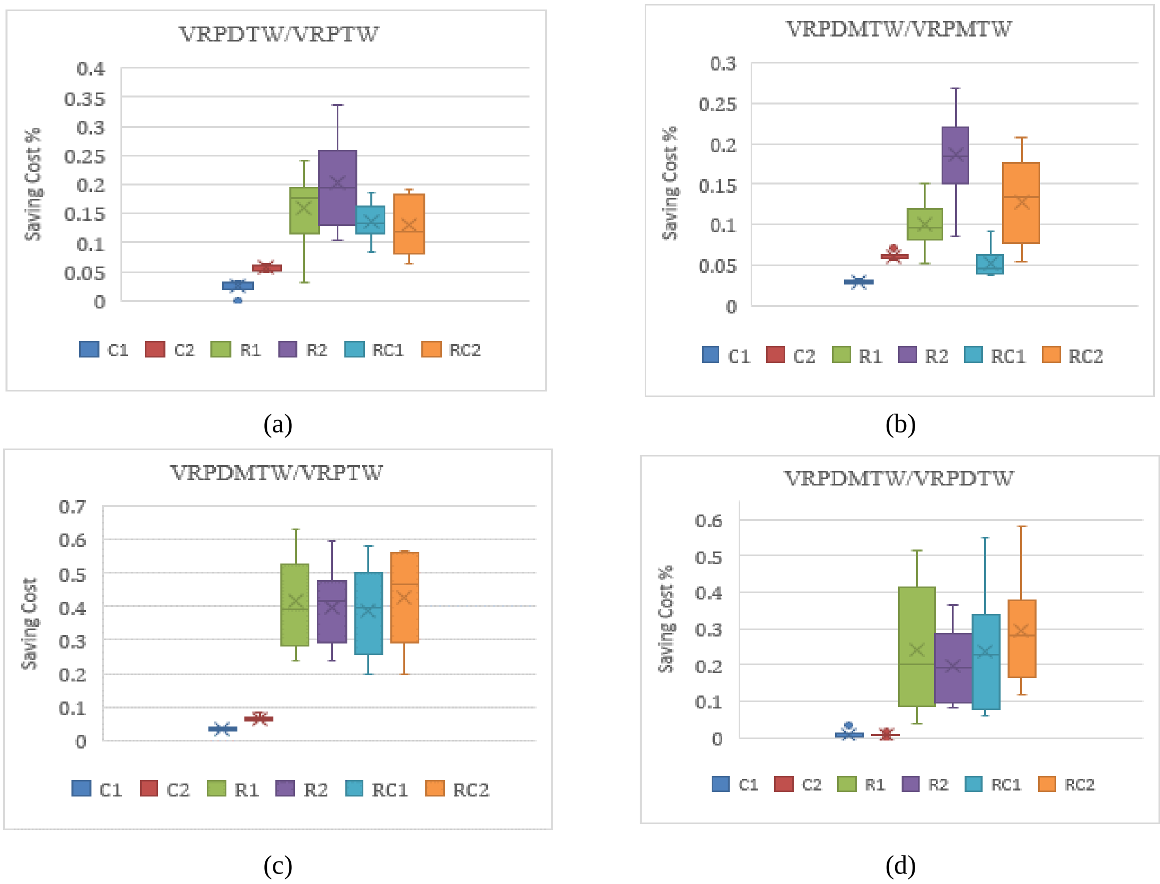


Figure 4.3. The potential savings obtained by comparing different routing platforms for different network configurations

4.5.4.2. Impact of the width and the density of time windows

The study analyses the impact of time window width and the percentage of customers selecting multiple time windows on costs. However, not all customers may be able to select multiple time slots for their deliveries. The study generates 105 test instances with varying time window widths and the percentage of clients willing to pick various time slots. We use the same method used by Solomon (1987) to generate network instances, where CC, RR, and RC represent the clustered, randomly distributed, and clustered-random distributions of consumers within the network, respectively. The results are reported in Table 4.6 and Figure 4.4, which shows the average objective values for different network configurations and time window widths based on the percentage of customers selecting multiple time windows.

Widening the time windows results in an overall cost reduction across all networks, as seen in Figure 4.6. For instance, by altering the width of the time windows from 1 to 5, the objective values in network configuration CC fall from 1400 to 800. Additionally, as more consumers choose multiple time windows, there is a noticeable cost decrease across all networks. In network CC, for instance, the objective value falls from 1400 to 600, even though the time window width is set to 1.

Table 4.6. Impact of the width and portion of customers who opt for multiple time windows. The column Z^{ALNS} represents the best objective obtained by the ALNS. ΔTW represents the width of the time windows. The portion of customers opting for multiple time windows is indicated by TW%. CPU shows the average execution time of the ALNS algorithm for each instance. $T - customer$ and $D - customer$ are the number of customers served by the trucks and drones, respectively. NT and ND are the number of trucks and drones used, respectively.

Ins.	ΔTW	$TW\%$	Z^{ALNS}	$CPU(S)$	$T - customer$	$D - customer$	NT	ND
CC	1	0	1364.00	20.98	76	24	7	4
CC	1	20	1067.70	18.13	75	25	8	4
CC	1	40	743.01	28.69	75	25	2	4
CC	1	50	621.90	35.67	66	34	2	4
CC	1	60	609.19	30.89	81	19	2	3
CC	1	80	593.39	42.29	69	31	2	4
CC	1	100	575.55	36.39	67	33	2	4
Mean			796.39	30.43	72.71	27.29	3.57	3.86
CC	2	0	1254.97	18.56	77	23	15	4
CC	2	20	928.96	18.81	70	30	6	5
CC	2	40	649.18	34.59	70	30	2	4
CC	2	50	608.24	27.42	75	25	2	4
CC	2	60	602.38	38.52	65	35	2	4
CC	2	80	575.55	36.39	67	33	2	4
CC	2	100	551.29	29.40	76	24	2	4
Mean			738.65	29.10	71.43	28.57	4.43	4.14
CC	3	0	1180.00	35.44	71	29	2	4
CC	3	20	652.64	19.39	78	22	8	3
CC	3	40	625.48	30.28	68	32	2	4
CC	3	50	606.77	20.21	73	27	3	6
CC	3	60	596.58	18.72	68	32	3	6
CC	3	80	546.54	41.25	74	26	2	4
CC	3	100	530.59	29.75	66	34	3	6
Mean			676.94	27.86	71.14	28.86	3.29	4.71
CC	4	0	1133.25	21.87	76	24	5	4
CC	4	20	652.55	26.71	76	24	2	4
CC	4	40	566.46	19.14	63	37	2	4
CC	4	50	543.65	23.29	66	34	3	6
CC	4	60	541.23	18.61	79	21	3	5
CC	4	80	534.17	22.69	75	25	3	6

CC	4	100	528.91	28.92	67	33	3	6
Mean			642.89	23.03	71.71	28.29	3.00	5.00
CC	5	0	791.17	16.36	83	17	3	4
CC	5	20	630.57	37.82	73	27	2	4
CC	5	40	537.05	26.29	72	28	3	6
CC	5	50	533.53	20.68	73	27	3	5
CC	5	60	533.23	25.43	70	30	3	6
CC	5	80	532.58	25.71	63	37	3	6
CC	5	100	527.91	22.75	65	35	2	4
Mean			583.72	25.00	71.29	28.71	2.71	5.00
RR	1	0	1840.79	19.54	87	13	12	4
RR	1	20	1665.60	16.19	89	11	5	4
RR	1	40	1654.24	16.60	89	11	6	3
RR	1	50	1592.57	18.53	82	18	6	5
RR	1	60	1539.78	18.02	90	10	6	4
RR	1	80	1282.62	17.34	87	13	4	4
RR	1	100	890.85	27.07	74	26	5	8
Mean			1495.21	19.04	85.43	14.57	6.29	4.57
RR	2	0	1680.05	23.50	88	12	6	7
RR	2	20	1630.68	20.28	84	16	6	7
RR	2	50	1623.89	23.97	81	19	6	7
RR	2	40	1385.85	19.35	79	21	7	7
RR	2	60	918.67	34.31	87	13	4	6
RR	2	80	899.51	36.74	84	16	2	4
RR	2	100	879.74	40.68	75	25	3	4
Mean			1288.34	28.41	82.57	17.43	4.86	6.00
RR	3	0	1673.70	19.46	81	19	7	9
RR	3	20	1377.48	19.59	83	17	4	6
RR	3	40	1234.73	20.00	87	13	4	6
RR	3	50	981.34	27.08	89	11	2	3
RR	3	60	909.66	26.61	91	9	2	2
RR	3	80	890.29	31.63	79	21	5	9
RR	3	100	807.81	30.50	81	19	3	6
Mean			1125.00	24.98	84.43	15.57	3.86	5.86
RR	4	0	1519.07	19.82	85	15	5	3
RR	4	20	1301.08	18.28	88	12	4	3
RR	4	40	1160.91	17.83	79	21	4	4
RR	4	50	968.48	31.11	86	14	4	5
RR	4	60	866.46	25.02	81	19	3	6
RR	4	80	857.76	30.86	76	24	2	4

RR	4	100	750.98	33.82	73	27	2	4
Mean			1060.68	25.25	81.14	18.86	3.43	4.14
RR	5	0	1405.25	16.70	94	6	4	3
RR	5	20	897.95	27.61	79	21	5	9
RR	5	40	897.77	23.49	67	33	5	9
RR	5	50	868.62	31.34	80	20	3	6
RR	5	60	787.00	28.78	72	28	4	8
RR	5	80	782.10	28.53	80	19	3	5
RR	5	100	721.15	28.45	75	25	4	8
Mean			908.55	26.41	78.14	21.71	4.00	6.86
RC	1	0	2142.71	17.77	85	15	14	4
RC	1	20	1461.10	21.37	87	13	12	2
RC	1	40	1420.52	17.87	85	15	14	4
RC	1	50	999.16	44.19	81	19	3	4
RC	1	60	826.92	43.85	78	22	2	3
RC	1	80	812.60	43.63	73	27	2	4
RC	1	100	734.08	40.35	79	21	2	3
Mean			1199.59	32.72	81.14	18.86	7.00	3.43
RC	2	0	1869.60	17.76	81	19	18	4
RC	2	20	1353.03	25.52	88	12	8	2
RC	2	40	1125.63	24.15	83	17	7	3
RC	2	50	767.94	36.12	80	20	2	3
RC	2	60	754.22	24.94	74	26	3	5
RC	2	80	733.24	25.15	72	28	2	4
RC	2	100	716.56	21.87	71	29	2	4
Mean			1045.75	25.07	78.43	21.57	6.00	3.57
RC	3	0	1569.62	22.10	85	15	16	3
RC	3	20	1165.34	23.61	75	25	9	4
RC	3	40	1067.90	20.21	80	20	7	4
RC	3	50	752.82	28.11	69	31	2	4
RC	3	60	751.13	31.82	72	28	2	4
RC	3	80	718.76	33.24	75	25	3	5
RC	3	100	708.88	26.48	75	25	3	5
Mean			962.06	26.51	75.86	24.14	6.00	4.14
RC	4	0	1348.81	15.82	85	15	4	4
RC	4	20	770.36	21.44	91	9	2	2
RC	4	40	763.76	24.80	77	23	2	4
RC	4	50	743.53	22.49	72	28	2	4
RC	4	60	739.80	29.38	71	29	2	4
RC	4	80	708.12	31.49	72	28	2	4

RC	4	100	704.93	30.15	80	20	2	3
Mean			825.62	25.08	78.29	21.71	2.29	3.57
RC	5	0	1125.17	18.60	74	26	5	3
RC	5	20	739.38	33.21	81	19	2	3
RC	5	40	736.39	20.35	70	30	2	3
RC	5	50	735.66	26.01	71	29	2	4
RC	5	60	724.36	60.19	67	33	2	4
RC	5	80	699.54	30.60	73	27	2	4
RC	5	100	698.61	26.52	73	27	2	3
Mean			779.87	30.78	72.71	27.29	2.43	3.43

Figure 4.4 shows that as more customers select multiple time windows, the variation in objective value caused by adjusting the time window width reduces. This suggests that a more flexible delivery system offering multiple time windows can prevent a major cost increase even with narrow time windows. Therefore, grocery providers can provide narrower time windows if they can convince customers to select multiple time slots, which can improve customer service (Narrowing time windows reduces waiting time for customers.).

This conclusion makes greater sense for network distributions that are clustered (CC) or random-clustered (RC). The change in objective values with wide and narrow time windows is not that substantial, though, when the fraction of consumers choosing various time windows surpasses 60%, as seen in Figure 4.4. This result is also generalized to randomly distributed networks (RR).

4.6 CONCLUSION

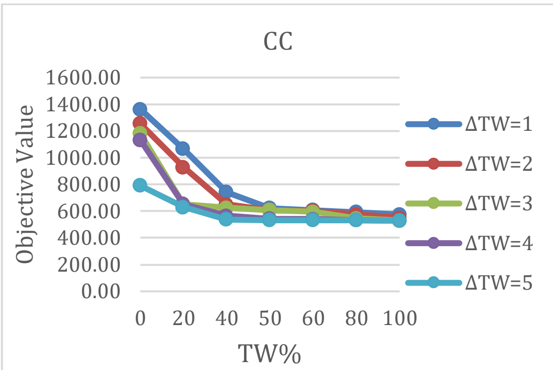
This chapter proposed a MIP model for the home delivery of e-groceries using a vehicle-drone routing problem with multiple time windows. The VRPDMTW model is a novel variation of VRPD that enables customers to choose multiple time windows. The proposed concept involves deliveries made cooperatively by trucks and drones, with synchronization constraints at rendezvous points. The routing strategy involves sending trucks with drones to customer locations and then moving to other stops to either service a customer or retrieve a drone.

A hybrid approach combining ALNS and the Constraint Programming (CP) model was devised, and its performance was assessed by a number of tests, as the standard solvers cannot provide solutions for large network instances. The benchmark instances provided by Solomon (1987) were used to assess the effectiveness of the suggested solution approach.

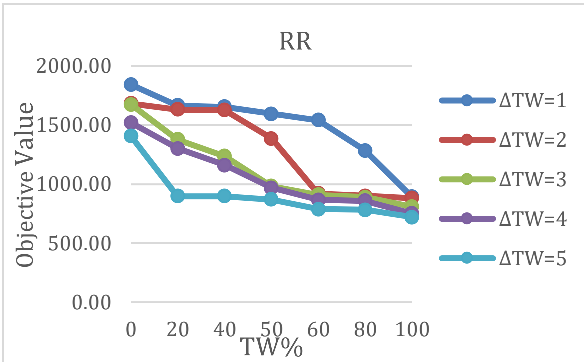
The numerical results demonstrated the algorithm's effectiveness since it can provide solutions with a less-than-0.15% optimality gap within a few seconds of execution.

A novel time window generation technique was implemented to generate test instances, and the potential savings were calculated to analyse the advantage of establishing the VRPDMTW. Three different network configurations—random, clustered, and random-clustered distribution networks—were tested numerically.

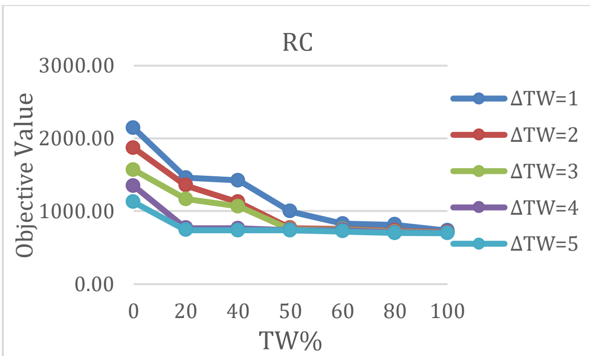
Results revealed that on average savings of approximately 29%, 16%, and 9% can be made by the VRPDMTW compared to the VRPTW, the VRPDTW, and the VRPMTW, respectively. The results suggested that the saving realized by introducing drones into delivery operations with multiple time windows is significant. The greatest variety of variances were seen in instances with randomly distributed customers. Additionally, the greatest savings were realized by these networks. In contrast, the clustered customers displayed the smallest cost reductions and variability.



(a)



(b)



(c)

Figure 4.4. Impact of the width and the percentage of customers choosing multiple time windows

Finally, we ran a sensitivity analysis on 105 test instances with varied time window widths and the proportion of customers that are willing to choose multiple time windows. According to the analysis, as the proportion of customers choosing multiple time slots increases, the change in objective achieved by varying the time window's width decreases. Hence if grocery providers can encourage clients to choose multiple time slots, they will be able to meet tighter time windows and so enhance the customer service.

Although attempts were made to present a comprehensive study, further real-world assumptions can make the model more practical. As a future study, split deliveries will be investigated as these would make a lot of sense in the e-grocery context. Considering energy consumption would also improve the current study further. Besides, an efficient algorithm to mitigate the computational effort to solve large network instances is always welcomed.

Chapter 5: Concluding Discussion

5.1 SUMMARY OF THE CONTRIBUTIONS AND FINDINGS

The aim of chapter 2 was to investigate a collaborative human-robot system for simultaneous delivery and pickup tasks. In this system, couriers serve end customers, while Autonomous Mobile Robots (AMLs) supply couriers with deliveries and collect pickups from couriers. To model this problem, a novel Mixed Integer Linear Programming (MILP) formulation was proposed. The MILP model includes a two-echelon network of multiple depots, stops, *m*-AMLs, and *n*-couriers. Couriers can perform multiple open routes, subject to constraints on payload and total distance, while AMLs can perform single open routes. This is a relaxation of the closed route problem as closed routes are not excluded from the solution space. The objective function is to minimize total fixed and variable costs, where fixed costs relate to the number of AMLs and couriers deployed, and variable costs relate to the total distance covered by AMLs and couriers.

Since the MILP model cannot solve large-scale problems, a heuristic algorithm combining several heuristics was developed and tested using numerical experiments. The results showed that the algorithm provides optimal solutions for most small-sized instances. An experiment was conducted to examine the efficiency of the algorithm. The results indicated that the MILP solver can rarely improve the initial solution provided by the heuristic algorithm for medium-sized networks.

The proposed algorithm for solving large-scale instances was compared with three other variants of the simultaneous pickup and delivery problem. The results show that a significant cost savings can be achieved by using a robot-human cooperative system where couriers can perform multiple open routes. Finally, the impact of the capacity and maximum range of the AMLs on the total cost was analysed. The results showed that increasing the capacity from 300 to 700 kg in a network with a size of 6×6 km^2 and 150 customers can reduce the total cost by up to 30%. In the same network configuration, increasing the range of AMLs from 1 to 3 km results in a 16.8% improvement in the total cost. The sensitivity analysis showed that focusing on the use of AMLs with higher capacity has a greater effect on reducing costs than increasing their maximum range.

Chapter 3 introduced a delivery system utilizing trucks and drones, presenting two heuristics along with a Mixed Integer Programming (MIP) model. The model includes truck and drone operational parameters and customer locations. The solution process considers all truck-drone collection-dispatch strategies, including dispatch-waiting-collection or dispatch-move-collection scenarios. Items may be delivered to customers by truck or drone, with the assignment determined by the solution process. Drones are dispatched from trucks to deliver orders and fly towards customer locations. Drones must land at docking stations to rejoin the trucks, and they can join any truck, not just the one from which they were dispatched.

Due to the intractability of the vehicle routing problem, two efficient algorithms using heuristics were developed to solve the problem for large instances, and the efficiency of the algorithms was tested by numerical experiments. The results showed good performance of the heuristic algorithms in terms of the execution time and the quality of solutions.

Sensitivity analysis was performed on several parameters, which showed that in networks with different configurations and different population densities, the optimal ratio of flight range and load carrying capacity of drones can be different. For more densely populated networks, a higher carrying capacity for drones is more desirable, while for wider networks, flight range becomes more important.

In Chapter 4, a novel MIP model is proposed for delivering e-groceries to customers' homes. The model, known as the VRPDMTW, is an extension of the VRPD that allows customers to choose multiple time windows. The proposed approach involves a combination of truck and drone deliveries with synchronization constraints at rendezvous points. The routing strategy involves sending trucks with drones to customer locations and then moving to other stops to either service a customer or retrieve a drone.

To assess the effectiveness of the proposed approach, a hybrid ALNS and CP model was developed and tested as standard solvers were not suitable for large network instances. The benchmark instances provided by Solomon (1987) were used to generated various datasets and to evaluate the proposed approach. The numerical results showed that the algorithm can provide solutions with a less-than-0.15% optimality gap within a few seconds of execution.

A novel time window generation technique was used to generate test instances, and the potential savings were calculated to analyse the advantages of using the VRPDMTW. The numerical tests were carried out on three different network configurations; random, clustered, and random-clustered distribution networks.

The results showed that the VRPDMTW had average savings of approximately 29%, 16%, and 9% compared to the VRPTW, the VRPDTW, and the VRPMTW, respectively. Networks with randomly distributed customers displayed the greatest variety of variances and the greatest savings, while clustered customers displayed the smallest cost reductions and variability.

Sensitivity analysis was performed on 105 test instances with varied time window widths and the proportion of customers willing to choose multiple time windows. The analysis showed that as the proportion of customers choosing multiple time slots increases, the change in objective achieved by varying the time window's width decreases. Therefore, grocery providers should encourage clients to choose multiple time slots to enhance customer service and meet tighter time windows.

The main contributions of this thesis are listed in Section 5.2.

5.2 CONTRIBUTIONS

1- Development of a comprehensive framework for optimizing pickup and delivery operations in a two-echelon network with simultaneous pickup and delivery operations, which includes the following features:

- An Autonomous Mobile Locker (AML)-Courier routing problem which allows couriers perform multiple tours in the second echelon.
- Open tours are allowed at both echelons of the network allowing couriers to serve customers in different residential zones.
- A novel cost optimizing MILP model (2E-MOVRP) is formulated to address the 2-echelon pickup and delivery problem in which multiple courier tours and open routes are allowed.
- A novel adaptive matheuristic is developed to deal with large networks.

2- Development of a flexible cooperation system between drones and trucks for efficient delivery operations, which includes the following features:

- An *m*-drone, *n*-truck routing framework that allows the drones to be retrieved by any other nearby truck that is not necessarily its mother truck, leading to a more flexible system of cooperation between drones and trucks.
- Various drone dispatch and collection strategies including dispatch-wait-collect and dispatch-move-collect.

3- Development of a more generalized drone-truck routing problem with multiple time windows, which includes the following features:

- Multiple time windows for each customer are considered among which customers are allowed to choose one- or multiple-time windows.
- An exact approach formulated as a Constraint Programming model is developed to solve the sub-problem of finding the optimum scheduling decisions for a given truck/drone route.
- A novel cost optimizing MILP model is formulated to address the problem.
- An Adaptive Large Neighbourhood Search (ALNS) heuristic approach is devised to deal with the complexity of the drone-truck routing problem for large instances.

Overall, these contributions represent significant advancements in the field of delivery operations, offering more comprehensive and flexible solutions for optimizing the routing and scheduling of delivery vehicles and drones.

5.3 FUTURE RESEARCH DIRECTIONS

Based on the findings thus far, here are some potential future research directions:

Multi-objective optimization: The MIP models proposed in Chapters 2, 3 and 4 have a single objective of minimizing the total cost. However, in real-world scenarios, there are often multiple conflicting objectives such as minimizing cost and maximizing customer satisfaction. Therefore, future research can focus on developing multi-objective optimization models to balance different objectives and establish Pareto optimal decision frontiers.

Robust optimization: The proposed models in Chapters 2, 3, and 4 assume that all the parameters such as demand, travel time, and drone range are known with certainty. However, in practice, these parameters are uncertain and may vary. Therefore, future research can focus on developing robust optimization models that consider uncertainties and provide solutions that are robust to these uncertainties.

Real-time decision-making: The proposed models in Chapters 2, 3, and 4 are designed for offline optimization, where all the data is available in advance. This would typically be the case for next day delivery or pickup. However, in real-world scenarios, new information or demands may arise during the execution of the pickup and/or delivery operations. Therefore, future research can focus on developing real-time decision-making algorithms that can adapt to changing conditions and make optimal decisions on the fly.

Environmental sustainability: The proposed models in Chapters 2, 3, and 4 focus primarily on cost optimization. However, delivery operations also have a significant impact on the environment, and there is a growing need to reduce carbon emissions and promote sustainable delivery systems. Therefore, future research can focus on developing models that consider environmental factors and energy consumption to promote sustainable delivery operations.

Optimization and algorithm design: The proposed models in all three chapters use optimization techniques to solve delivery problems involving human-robot or truck-drone collaboration. However, more efficient and effective optimization techniques can be developed to solve larger and more complex delivery problems.

Real-world testing and validation: While the proposed models and algorithms in these chapters show promising results in numerical experiments, they have yet to be tested in real-world scenarios. Future research can focus on deploying and testing these models and algorithms in actual delivery systems, collecting empirical data, and validating their effectiveness and practicality in real-world settings.

Integration with other technologies: Human-robot and truck-drone collaboration are part of a larger trend of integrating robotics, automation, and artificial intelligence into delivery systems. Future research can focus on exploring the potential of integrating these technologies with other emerging technologies, such as blockchain, Internet of Things (IoT), and edge computing, to create more efficient, secure, and resilient delivery systems. However, it should be acknowledged that new technologies create new vulnerabilities, in particular cyber vulnerabilities, which also need to be considered.

References

Agatz, N., Bouman, P., & Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, *52*(4), 965–981.

Ailawadi, K. L., & Farris, P. W. (2017). Managing multi-and omni-channel distribution: metrics and research directions. *Journal of Retailing*, *93*(1), 120–135.

Akhtar, M., Hannan, M. A., Begum, R. A., Basri, H., & Scavino, E. (2017). Backtracking search algorithm in CVRP models for efficient solid waste collection and route optimization. *Waste Management*, *61*, 117–128.

Alfandari, L., Ljubić, I., & da Silva, M. D. M. (2021). A tailored Benders decomposition approach for last-mile delivery with autonomous robots. *European Journal of Operational Research*.

Alfandari, L., Ljubić, I., & De Melo da Silva, M. (2022). A tailored Benders decomposition approach for last-mile delivery with autonomous robots. *European Journal of Operational Research*, *299*(2), 510–525. https://doi.org/10.1016/j.ejor.2021.06.048

Almuhaideb, S., Alhussan, T., Alamri, S., Altwaijry, Y., Aljarbou, L., & Alrayes, H. (2021). Optimization of truck-drone parcel delivery using metaheuristics. *Applied Sciences (Switzerland)*, *11*(14). https://doi.org/10.3390/app11146443

Al-Turjman, F., Lemayian, J. P., Alturjman, S., & Mostarda, L. (2019). Enhanced Deployment Strategy for the 5G Drone-BS Using Artificial Intelligence. *IEEE Access*, *7*, 75999–76008. https://doi.org/10.1109/ACCESS.2019.2921729

An, H. (2022). The closed-open truck and loader routing problem for biomass transportation from satellite storage locations to a bioenergy plant. *Computers and Electronics in Agriculture, 194*, 106764.

Bai, X., Cao, M., Yan, W., & Ge, S. S. (2020). Efficient Routing for Precedence-Constrained Package Delivery for Heterogeneous Vehicles. *IEEE Transactions on Automation Science and Engineering*, *17*(1), 248–260. https://doi.org/10.1109/TASE.2019.2914113

Bakach, I., Campbell, A. M., & Ehmke, J. F. (2021). A two-tier urban delivery network with robot-based deliveries. *Networks, 78*(4), 461–483. https://doi.org/10.1002/net.22024

Baptiste, P., Le Pape, C., & Nuijten, W. (2001). *Constraint-based scheduling: applying constraint programming to scheduling problems* (Vol. 39). Springer Science & Business Media.

Bartolini, E., & Schneider, M. (2020). A two-commodity flow formulation for the capacitated truck-and-trailer routing problem. *Discrete Applied Mathematics*, *275,* 3–18.

Beheshti, A. K., Hejazi, S. R., & Alinaghian, M. (2015). The vehicle routing problem with multiple prioritized time windows: A case study. *Computers & Industrial Engineering*, *90*, 402–413.

Belhaiza, S., Hansen, P., & Laporte, G. (2014). A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. *Computers & Operations Research*, *52*, 269–281.

Bergdoll, B., & Christensen, P. (2008). *Home delivery: fabricating the modern dwelling*. The Museum of Modern Art.

Bhattacharjya, J., Ellison, A., Tripathi, S. (2016). An exploration of logistics-related customer service provision on Twitter: The case of e-retailers. *International Journal of Physical Distribution & Logistics Management*, *46*, 659–680.

Boccia, M., Masone, A., Sforza, A., & Sterle, C. (2021). A column-and-row generation approach for the flying sidekick travelling salesman problem. *Transportation Research Part C: Emerging Technologies*, *124*(November 2020), 102913. https://doi.org/10.1016/j.trc.2020.102913

Boysen, N., Briskorn, D., Fedtke, S., & Schwerdfeger, S. (2018). Drone delivery from trucks: Drone scheduling for given truck routes. *Networks*, *72*(4), 506–527. https://doi.org/10.1002/net.21847

Boysen, N., Schwerdfeger, S., & Weidinger, F. (2018). Scheduling last-mile deliveries with truck-based autonomous robots. *European Journal of Operational Research*, *271*(3), 1085–1099. https://doi.org/10.1016/j.ejor.2018.05.058

Brandão, J. (2020). A memory-based iterated local search algorithm for the multi-depot open vehicle routing problem. *European Journal of Operational Research*, *284*(2), 559–571.

Bräysy, O., & Gendreau, M. (2005). Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science*, *39*(1), 104–118.

Cai, X., Jiang, L., Guo, S., Huang, H., & Du, H. (2021). TLHSA and SACA: two heuristic algorithms for two variant VRP models. *Journal of Combinatorial Optimization*. https://doi.org/10.1007/s10878-021-00831-0

Campbell, J. F., Corberán, Á., Plana, I., & Sanchis, J. M. (2018). Drone arc routing problems. *Networks*, *72*(4), 543–559. https://doi.org/10.1002/net.21858

Cavani, S., Iori, M., & Roberti, R. (2021). Exact methods for the traveling salesman problem with multiple drones. *Transportation Research Part C: Emerging Technologies*, *130*(November 2020), 103280. https://doi.org/10.1016/j.trc.2021.103280

Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, *45*(1), 41–51.

Chauhan, D., Unnikrishnan, A., & Figliozzi, M. (2019). Maximum coverage capacitated facility location problem with range constrained drones. *Transportation Research Part C: Emerging Technologies*, *99*(January), 1–18. https://doi.org/10.1016/j.trc.2018.12.001

Chen, C., Demir, E., & Huang, Y. (2021). An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and delivery robots. *European Journal of Operational Research*, *294*(3), 1164–1180. https://doi.org/10.1016/j.ejor.2021.02.027

Chen, C., Demir, E., Huang, Y., & Qiu, R. (2021). The adoption of self-driving delivery robots in last mile logistics. *Transportation Research Part E: Logistics and Transportation Review*, *146*(December 2020), 102214. https://doi.org/10.1016/j.tre.2020.102214

Chen, X., Ulmer, M. W., & Thomas, B. W. (2022). Deep Q-learning for same-day delivery with vehicles and drones. *European Journal of Operational Research*, *298*(3), 939–952. https://doi.org/10.1016/j.ejor.2021.06.021

Cheng, C., Adulyasak, Y., & Rousseau, L. M. (2020). Drone routing with energy function: Formulation and exact algorithm. *Transportation Research Part B: Methodological*, *139*, 364–387. https://doi.org/10.1016/j.trb.2020.06.011

Cheung, K. F., Bell, M. G. H., & Bhattacharjya, J. (2021). Cybersecurity in logistics and supply chain management: An overview and future research directions. *Transportation Research Part E: Logistics and Transportation Review*, *146*(July 2020), 102217. https://doi.org/10.1016/j.tre.2020.102217

Choudhury, S., Gupta, J. K., Kochenderfer, M. J., Sadigh, D., & Bohg, J. (2022). Dynamic multi-robot task allocation under uncertainty and temporal constraints. *Autonomous Robots*, *46*(1), 231–247. https://doi.org/10.1007/s10514-021-10022-9

Chung, S.-H. (2021). Applications of smart technologies in logistics and transport: A review. *Transportation Research Part E: Logistics and Transportation Review*, *153*, 102455.

Civicioglu, P. (2013). Backtracking search optimization algorithm for numerical optimization problems. *Applied Mathematics and Computation*, *219*(15), 8121–8144.

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

Coelho, B. N., Coelho, V. N., Coelho, I. M., Ochi, L. S., Haghnazar K., R., Zuidema, D., Lima, M. S. F., & da Costa, A. R. (2017). A multi-objective green UAV routing problem. *Computers and Operations Research*, *88*, 306–315. https://doi.org/10.1016/j.cor.2017.04.011

Coindreau, M. A., Gallay, O., & Zufferey, N. (2021). Parcel delivery cost minimization with time window constraints using trucks and drones. *Networks*, *78*(4), 400–420. https://doi.org/10.1002/net.22019

Coindreau, M., Gallay, O., & Zufferey, N. (2021). Parcel delivery cost minimization with time window constraints using trucks and drones. *Networks*, *78*(4), 400–420.

Cokyasar, T., Dong, W., Jin, M., & Verbas, İ. Ö. (2021). Designing a drone delivery network with automated battery swapping machines. *Computers and Operations Research*, *129*. https://doi.org/10.1016/j.cor.2020.105177

Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y., & Semet, F. (2002). A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, *53*(5), 512–522.

Dang, Q. V., Singh, N., Adan, I., Martagan, T., & van de Sande, D. (2021). Scheduling heterogeneous multi-load AGVs with battery constraints. *Computers and Operations Research*, *136*(October 2020), 105517. https://doi.org/10.1016/j.cor.2021.105517

Das, D. N., Sewani, R., Wang, J., & Tiwari, M. K. (2021). Synchronized Truck and Drone Routing in Package Delivery Logistics. *IEEE Transactions on Intelligent Transportation Systems*, *22*(9), 5772–5782. https://doi.org/10.1109/TITS.2020.2992549

Dayarian, I., Savelsbergh, M., & Clarke, J. P. (2020). Same-day delivery with drone resupply. *Transportation Science*, *54*(1), 229–249. https://doi.org/10.1287/trsc.2019.0944

Dell'Amico, M., Montemanni, R., & Novellani, S. (2021a). Drone-assisted deliveries: new formulations for the flying sidekick traveling salesman problem. In *Optimization Letters* (Vol. 15, Issue 5). Springer Berlin Heidelberg. https://doi.org/10.1007/s11590-019-01492-z

Dell'Amico, M., Montemanni, R., & Novellani, S. (2021b). Modeling the flying sidekick traveling salesman problem with multiple drones. *Networks*, *78*(3), 303–327. https://doi.org/10.1002/net.22022

Demir, E., Bektaş, T., & Laporte, G. (2012). An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, *223*(2), 346–359.

Di Puglia Pugliese, L., & Guerriero, F. (2017). Last-mile deliveries by using drones and classical vehicles. *International Conference on Optimization and Decision Science*, 557–565.

Di Puglia Pugliese, L., Macrina, G., & Guerriero, F. (2021). Trucks and drones cooperation in the last-mile delivery process. *Networks*, *78*(4), 371–399. https://doi.org/10.1002/net.22015

Dorling, K., Heinrichs, J., Messier, G. G., & Magierowski, S. (2016). Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *47*(1), 70–85.

Dorling, K., Heinrichs, J., Messier, G. G., & Magierowski, S. (2017). Vehicle Routing Problems for Drone Delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *47*(1), 70–85. https://doi.org/10.1109/TSMC.2016.2582745

Elsayed, M., & Mohamed, M. (2020). The impact of airspace regulations on unmanned aerial vehicles in last-mile operation. *Transportation Research Part D: Transport and Environment*, *87*(August), 102480. https://doi.org/10.1016/j.trd.2020.102480

Favaretto, D., Moretti, E., & Pellegrini, P. (2007). Ant colony system for a VRP with multiple time windows and multiple visits. *Journal of Interdisciplinary Mathematics*, *10*(2), 263–284.

Ferrandez, S. M., Harbison, T., Weber, T., Sturges, R., & Rich, R. (2016). Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. *Journal of Industrial Engineering and Management (JIEM)*, *9*(2), 374–388.

Fleischmann, B. (1990). The vehicle routing problem with multiple use of vehicles. *Forschungsbericht Fachbereich Wirtschaftswissenschaften, Universität Hamburg*.

François, V., Arda, Y., Crama, Y., & Laporte, G. (2016). *Large neighborhood search for multi-trip vehicle routing*. *255*, 422–441. https://doi.org/10.1016/j.ejor.2016.04.065

Freitas, J. C. de, & Penna, P. H. V. (2020). A variable neighborhood search for flying sidekick traveling salesman problem. *International Transactions in Operational Research*, *27*(1), 267–290. https://doi.org/10.1111/itor.12671

Gasoline prices around the world, 02-Nov-2020 | GlobalPetrolPrices.com. (n.d.). Retrieved November 9, 2020, from https://www.globalpetrolprices.com/gasoline_prices/

Ghelichi, Z., Gentili, M., & Mirchandani, P. B. (2021). Logistics for a fleet of drones for medical item delivery: A case study for Louisville, KY. *Computers and Operations Research*, *135*(June), 105443. https://doi.org/10.1016/j.cor.2021.105443

Gomez-Lagos, J., Candia-Vejar, A., & Encina, F. (2021). A New Truck-Drone Routing Problem for Parcel Delivery Services Aided by Parking Lots. *IEEE Access*, *9*, 11091–11108. https://doi.org/10.1109/ACCESS.2021.3050658

Gonzalez-R, P. L., Canca, D., Andrade-Pineda, J. L., Calle, M., & Leon-Blanco, J. M. (2020). Truck-drone team logistics: A heuristic approach to multi-drop route planning. *Transportation Research Part C: Emerging Technologies*, *114*(February), 657–680. https://doi.org/10.1016/j.trc.2020.02.030

Grangier, P., Gendreau, M., Lehuédé, F., & Rousseau, L.-M. (2016). An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. *European Journal of Operational Research*, *254*(1), 80–91.

Gu, Y., & Wallace, S. W. (2021). Operational benefits of autonomous vessels in logistics—A case of autonomous water-taxis in Bergen. *Transportation Research Part E: Logistics and Transportation Review*, *154*(December 2020), 102456. https://doi.org/10.1016/j.tre.2021.102456

Ha, Q. M., Deville, Y., Pham, Q. D., & Hà, M. H. (2018). On the min-cost Traveling Salesman Problem with Drone. *Transportation Research Part C: Emerging Technologies*, *86*(November 2017), 597–621. https://doi.org/10.1016/j.trc.2017.11.015

Ha, Q. M., Deville, Y., Pham, Q. D., & Ha, M. H. (2018). On the min-cost traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies*, *86*, 597–621.

Ham, A. M. (2018a). Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. *Transportation Research Part C: Emerging Technologies*, *91*(April), 1–14. https://doi.org/10.1016/j.trc.2018.03.025

Ham, A. M. (2018b). Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. *Transportation Research Part C: Emerging Technologies*, *91*, 1–14.

Hamzeei, M., Farahani, R. Z., & Rashidi-Bejgan, H. (2013). An exact and a simulated annealing algorithm for simultaneously determining flow path and the location of P/D stations in bidirectional path. *Journal of Manufacturing Systems*, *32*(4), 648–654. https://doi.org/10.1016/j.jmsy.2013.07.002

Han, Y. Q., Li, J. Q., Liu, Z., Liu, C., & Tian, J. (2020). Metaheuristic algorithm for solving the multi-objective vehicle routing problem with time window and drones. *International Journal of Advanced Robotic Systems*, *17*(2), 1–14. https://doi.org/10.1177/1729881420920031

Hasan, H. H., & Rahim, S. A. (2008). Factors affecting online purchasing behavior. *Jurnal Komunikasi, Malaysian Journal of Communication*, *24*(1), 1–19.

Hemmelmayr, V. C., Cordeau, J.-F., & Crainic, T. G. (2012). An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & Operations Research*, *39*(12), 3215–3228.

Hernández, E. J. U., Martínez, J. A. S., & Saucedo, J. A. M. (2020). Optimization of the distribution network using an emerging technology. *Applied Sciences (Switzerland)*, *10*(3). https://doi.org/10.3390/app10030857

Hong, I., Kuby, M., & Murray, A. T. (2018). A range-restricted recharging station coverage model for drone delivery service planning. *Transportation Research Part C: Emerging Technologies*, *90*(October 2017), 198–212. https://doi.org/10.1016/j.trc.2018.02.017

Hoogeboom, M., Dullaert, W., Lai, D., & Vigo, D. (2020). Efficient neighborhood evaluations for the vehicle routing problem with multiple time windows. *Transportation Science*, *54*(2), 400–416.

Huang, H., Savkin, A. V., & Huang, C. (2020). A New Parcel Delivery System with Drones and a Public Train. *Journal of Intelligent and Robotic Systems: Theory and Applications*, *100*(3–4), 1341–1354. https://doi.org/10.1007/s10846-020-01223-y

Huang, S. H., Huang, Y. H., Blazquez, C. A., & Chen, C. Y. (2022). Solving the vehicle routing problem with drone for delivery services using an ant colony optimization algorithm. *Advanced Engineering Informatics*, *51*(October 2021), 101536. https://doi.org/10.1016/j.aei.2022.101536

Hübner, A. H., Kuhn, H., Wollenburg, J., Towers, N., & Kotzab, H. (2016). Last mile fulfilment and distribution in omni-channel grocery retailing: a strategic planning framework. *International Journal of Retail & Distribution Management*, *44*(3), 228–247.

Iranmanesh, S., Raad, R., Raheel, M. S., Tubbal, F., & Jan, T. (2020). Novel DTN Mobility-Driven Routing in Autonomous Drone Logistics Networks. *IEEE Access*, *8*, 13661–13673. https://doi.org/10.1109/ACCESS.2019.2959275

Ishfaq, R., Defee, C., Gibson, B.J. and Raja, U. (2016). Realignment of the physical distribution process in omni-channel fulfilment. *International Journal of Physical Distribution & Logistics Management*, *46 Nos 6/7*, 543–561.

Jayaraman, V., & Ross, A. (2003). A simulated annealing methodology to distribution network design and management. *European Journal of Operational Research*, *144*(3), 629–645.

Jeon, A., Kang, J., Choi, B., Kim, N., Eun, J., & Cheong, T. (2021). Unmanned Aerial Vehicle Last-Mile Delivery Considering Backhauls. *IEEE Access*, *9*, 85017–85033. https://doi.org/10.1109/ACCESS.2021.3087751

Joerss, M., Schröder, J., Neuhaus, F., Klink, C., & Mann, F. (2016). Parcel delivery: The future of last mile. *McKinsey & Company*, 1–32.

Jun, S., Lee, S., & Yih, Y. (2021). Pickup and delivery problem with recharging for material handling systems utilising autonomous mobile robots. *European Journal of Operational Research*, *289*(3), 1153–1168. https://doi.org/10.1016/j.ejor.2020.07.049

Jung, H., & Kim, J. (2022). Drone scheduling model for delivering small parcels to remote islands considering wind direction and speed. *Computers and Industrial Engineering*, *163*(November 2021), 107784. https://doi.org/10.1016/j.cie.2021.107784

Kang, M., & Lee, C. (2021). An exact algorithm for heterogeneous drone-truck routing problem. *Transportation Science*, *55*(5), 1088–1112.

Karak, A., & Abdelghany, K. (2019). The hybrid vehicle-drone routing problem for pick-up and delivery services. *Transportation Research Part C: Emerging Technologies*, *102*(September 2018), 427–449. https://doi.org/10.1016/j.trc.2019.03.021

Kaspi, M., Kesselman, U., & Tanchoco, J. M. A. (2002). Optimal solution for the flow path design problem of a balanced unidirectional AGV system. *International Journal of Production Research*, *40*(2), 389–401. https://doi.org/10.1080/00207540110079761

Kim 2016. (n.d.). *The most staggering part about Amazon's upcoming drone delivery service | Business Insider*. Retrieved November 9, 2020, from https://www.businessinsider.com.au/cost-savings-from-amazon-drone-deliveries-2016-6?r=US&IR=T

Kim, J., Moon, H., & Jung, H. (2020). Drone-based parcel delivery using the rooftops of city buildings: Model and solution. *Applied Sciences (Switzerland)*, *10*(12). https://doi.org/10.3390/app10124362

Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1987). Optimization by simulated annealing. In *Readings in Computer Vision* (pp. 606–615). Elsevier.

Kitjacharoenchai, P., Min, B. C., & Lee, S. (2020). Two echelon vehicle routing problem with drones in last mile delivery. *International Journal of Production Economics*, *225*(December 2019), 107598. https://doi.org/10.1016/j.ijpe.2019.107598

Kitjacharoenchai, P., Min, B.-C., & Lee, S. (2019). Two echelon vehicle routing problem with drones in last mile delivery. *International Journal of Production Economics*, 107598.

Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J. M. A., & Brunese, P. A. (2019). Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. *Computers & Industrial Engineering*, *129*, 14–30.

Koster, K. L. (2015). *Delivery platform for unmanned aerial vehicles*. Google Patents.

Kuo, R. J., Lu, S.-H., Lai, P.-Y., & Mara, S. T. W. (2022). Vehicle routing problem with drones considering time windows. *Expert Systems with Applications*, *191*, 116264.

Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, *59*(3), 345–358.

Leon-Blanco, J. M., Gonzalez-R, P. L., Andrade-Pineda, J. L., Canca, D., & Calle, M. (2022). A multi-agent approach to the truck multi-drone routing problem. *Expert Systems with Applications*, *195*, 116604. https://doi.org/10.1016/j.eswa.2022.116604

Li, F., Golden, B., & Wasil, E. (2007). The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. *Computers & Operations Research*, *34*(10), 2918–2930.

Li, H., Chen, J., Wang, F., & Zhao, Y. (2022). Truck and drone routing problem with synchronization on arcs. *Naval Research Logistics*, *May 2021*, 1–18. https://doi.org/10.1002/nav.22053

Li, J., Ensafian, H., Bell, M. G. H., & Geers, D. G. (2021). Deploying autonomous mobile lockers in a two-echelon parcel operation. *Transportation Research Part C: Emerging Technologies*, *128*(December 2020), 103155. https://doi.org/10.1016/j.trc.2021.103155

Li, J., Rombaut, E., & Vanhaverbeke, L. (2021). A systematic review of agent-based models for autonomous vehicles in urban mobility and logistics: Possibilities for integrated simulation models. *Computers, Environment and Urban Systems*, *89*(July), 101686. https://doi.org/10.1016/j.compenvurbsys.2021.101686

Lin, C., Choy, K. L., Ho, G. T. S., Chung, S. H., & Lam, H. Y. (2014). Survey of Green Vehicle Routing Problem: Past and future trends. *Expert Systems with Applications*, *41*(4 PART 1), 1118–1138. https://doi.org/10.1016/j.eswa.2013.07.107

Lin, M., Lyu, J. Y., Gao, J. J., & Li, L. Y. (2020). Model and Hybrid Algorithm of Collaborative Distribution System with Multiple Drones and a Truck. *Scientific Programming*, *2020*. https://doi.org/10.1155/2020/8887057

Lin, S.-W., Vincent, F. Y., & Chou, S.-Y. (2009). Solving the truck and trailer routing problem based on a simulated annealing heuristic. *Computers & Operations Research*, *36*(5), 1683–1692.

Liu, D., Deng, Z., Mao, X., Yang, Y., & Kaisar, E. I. (2020). Two-Echelon vehicle-routing problem: optimization of autonomous delivery vehicle-assisted E-grocery distribution. *IEEE Access*, *8*, 108705–108719.

Liu, D., Deng, Z., Zhang, W., Wang, Y., & Kaisar, E. I. (2021). Design of sustainable urban electronic grocery distribution network. *Alexandria Engineering Journal*, *60*(1), 145–157. https://doi.org/10.1016/j.aej.2020.06.051

Liu, D., Yan, P., Pu, Z., Wang, Y., & Kaisar, E. I. (2021). Hybrid artificial immune algorithm for optimizing a Van-Robot E-grocery delivery system. *Transportation Research Part E: Logistics and Transportation Review*, *154*(November 2020), 102466. https://doi.org/10.1016/j.tre.2021.102466

Loske, D. (2020). The impact of COVID-19 on transport volume and freight capacity dynamics: An empirical analysis in German food retail logistics. *Transportation Research Interdisciplinary Perspectives*, *6*, 100165.

Luo, H., Zhang, P., Wang, J., Wang, G., & Meng, F. (2019). Traffic patrolling routing problem with drones in an urban road system. *Sensors (Switzerland)*, *19*(23), 1–20. https://doi.org/10.3390/s19235164

Luo, Z., Poon, M., Zhang, Z., Liu, Z., & Lim, A. (2021). The Multi-visit Traveling Salesman Problem with Multi-Drones. *Transportation Research Part C: Emerging Technologies*, *128*(September 2020), 103172. https://doi.org/10.1016/j.trc.2021.103172

Making online grocery a winning proposition | McKinsey. (n.d.). Retrieved June 26, 2023, from https://www.mckinsey.com/industries/consumer-packaged-goods/our-insights/making-online-grocery-a-winning-proposition

Mara, S. T. W., Rifai, A. P., & Sopha, B. M. (2022). An adaptive large neighborhood search heuristic for the flying sidekick traveling salesman problem with multiple drops. *Expert Systems with Applications*, 117647.

Marina Pasquali. (2023). *E-commerce worldwide - statistics & facts*.

Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, *26*(6), 369–395. https://doi.org/10.1007/s00158-003-0368-6

Masone, A., Poikonen, S., & Golden, B. L. (2022). The multivisit drone routing problem with edge launches: An iterative approach with discrete and continuous improvements. *Networks*, *December 2021*, 1–23. https://doi.org/10.1002/net.22087

Mbiadou Saleu, R. G., Deroussi, L., Feillet, D., Grangeon, N., & Quilliot, A. (2018). An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem. *Networks*, *72*(4), 459–474. https://doi.org/10.1002/net.21846

Mbiadou Saleu, R. G., Deroussi, L., Feillet, D., Grangeon, N., & Quilliot, A. (2022). The Parallel Drone Scheduling Problem with Multiple Drones and Vehicles. *European Journal of Operational Research*, *300*(2), 571–589. https://doi.org/10.1016/j.ejor.2021.08.014

Mesjasz-Lech, A., & Włodarczyk, A. (2022). The role of logistics infrastructure in development of sustainable road transport in Poland. *Research in Transportation Business and Management*, *xxxx*, 100841. https://doi.org/10.1016/j.rtbm.2022.100841

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, *21*(6), 1087–1092.

Momeni, M., Soleimani, H., Shahparvari, S., & Afshar-Nadjafi, B. (2022). Coordinated routing system for fire detection by patrolling trucks with drones. *International Journal of Disaster Risk Reduction*, *73*(January), 102859. https://doi.org/10.1016/j.ijdrr.2022.102859

Moshref-Javadi, M., Hemmati, A., & Winkenbach, M. (2020). A truck and drones model for last-mile delivery: A mathematical model and heuristic approach. *Applied Mathematical Modelling*, *80*, 290–318. https://doi.org/10.1016/j.apm.2019.11.020

Moshref-Javadi, M., Hemmati, A., & Winkenbach, M. (2021). A comparative analysis of synchronized truck-and-drone delivery models. *Computers and Industrial Engineering*, *162*(September 2020), 107648. https://doi.org/10.1016/j.cie.2021.107648

Moshref-Javadi, M., Lee, S., & Winkenbach, M. (2020). Design and evaluation of a multi-trip delivery model with truck and drones. *Transportation Research Part E: Logistics and Transportation Review*, *136*(April 2019), 101887. https://doi.org/10.1016/j.tre.2020.101887

Moshref-javadi, M., Lee, S., & Winkenbach, M. (2020). Design and evaluation of a multi-trip delivery model with truck and drones. *Transportation Research Part E*, *136*(February), 101887. https://doi.org/10.1016/j.tre.2020.101887

Moshref-Javadi, M., & Winkenbach, M. (2021). Applications and Research avenues for drone-based models in logistics: A classification and review. *Expert Systems with Applications*, *177*, 114854.

Mourad, A., Puchinger, J., & Van Woensel, T. (2021). Integrating autonomous delivery service into a passenger transportation system. *International Journal of Production Research*, *59*(7), 2116–2139. https://doi.org/10.1080/00207543.2020.1746850

Mualla, Y., Najjar, A., Daoud, A., Galland, S., Nicolle, C., Yasar, A. U. H., & Shakshuki, E. (2019). Agent-based simulation of unmanned aerial vehicles in civilian applications: A systematic literature review and research directions. *Future Generation Computer Systems*, *100*, 344–364. https://doi.org/10.1016/j.future.2019.04.051

Murray, C. C., & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, *54*, 86–109. https://doi.org/10.1016/j.trc.2015.03.005

Murray, C. C., & Raj, R. (2020). The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. *Transportation Research Part C: Emerging Technologies*, *110*(October 2019), 368–398. https://doi.org/10.1016/j.trc.2019.11.003

Nguyen, M. A., Dang, G. T. H., Hà, M. H., & Pham, M. T. (2022). The min-cost parallel drone scheduling vehicle routing problem. *European Journal of Operational Research*, *299*(3), 910–930. https://doi.org/10.1016/j.ejor.2021.07.008

Nucamendi-Guillén, S., Padilla, A. G., Olivares-Benitez, E., & Moreno-Vega, J. M. (2021). The multi-depot open location routing problem with a heterogeneous fixed fleet. *Expert Systems with Applications*, *165*, 113846.

Ostermeier, M., Heimfarth, A., & Hübner, A. (2021). Cost-optimal truck-and-robot routing for last-mile delivery. *Networks*.

Ostermeier, M., Heimfarth, A., & Hübner, A. (2022). Cost-optimal truck-and-robot routing for last-mile delivery. *Networks*, *79*(3), 364–389. https://doi.org/10.1002/net.22030

Peng, K., Du, J., Lu, F., Sun, Q., Dong, Y., Zhou, P., & Hu, M. (2019). A Hybrid Genetic Algorithm on Routing and Scheduling for Vehicle-Assisted Multi-Drone Parcel Delivery. *IEEE Access*, *7*, 49191–49200. https://doi.org/10.1109/ACCESS.2019.2910134

Perrotta, D., Grow, A., Rampazzo, F., Cimentada, J., Del Fava, E., Gil-Clavel, S., & Zagheni, E. (2021). Behaviours and attitudes in response to the COVID-19 pandemic: insights from a cross-national Facebook survey. *EPJ Data Science, 10*(1), 17.

Phan, D. H., & Suzuki, J. (2016). Evolutionary Multiobjective Optimization for the Pickup and Delivery Problem with Time Windows and Demands. *Mobile Networks and Applications*, *21*(1), 175–190. https://doi.org/10.1007/s11036-016-0709-5

Pina-Pardo, J. C., Silva, D. F., & Smith, A. E. (2021). The traveling salesman problem with release dates and drone resupply. *Computers and Operations Research*, *129*, 105170. https://doi.org/10.1016/j.cor.2020.105170

Pinto, R., & Lagorio, A. (2022). Point-to-point drone-based delivery network design with intermediate charging stations. *Transportation Research Part C: Emerging Technologies*, *135*(December 2021), 103506. https://doi.org/10.1016/j.trc.2021.103506

Poikonen, S., & Golden, B. (2020). Multi-visit drone routing problem. *Computers and Operations Research*, *113*, 104802. https://doi.org/10.1016/j.cor.2019.104802

Ponza, A. (2016). *Optimization of drone-assisted parcel delivery*. tesi.cab.unipd.it

Puerto, J., & Valverde, C. (2022). Routing for unmanned aerial vehicles: Touring dimensional sets. *European Journal of Operational Research*, *298*(1), 118–136. https://doi.org/10.1016/j.ejor.2021.06.061

Raj, R., & Murray, C. (2020). The multiple flying sidekicks traveling salesman problem with variable drone speeds. *Transportation Research Part C: Emerging Technologies*, *120*(September), 102813. https://doi.org/10.1016/j.trc.2020.102813

Rashidzadeh, E., Hadji Molana, S. M., Soltani, R., & Hafezalkotob, A. (2021). Assessing the sustainability of using drone technology for last-mile delivery in a blood supply chain. *Journal of Modelling in Management*, *16*(4), 1376–1402. https://doi.org/10.1108/JM2-09-2020-0241

Reed, S., Campbell, A. M., & Thomas, B. W. (2022). The Value of Autonomous Vehicles for Last-Mile Deliveries in Urban Environments. *Management Science*, *68*(1), 280–299. https://doi.org/10.1287/mnsc.2020.3917

Ren, X., Vashisht, S., Aujla, G. S., & Zhang, P. (2022). Drone-Edge Coalesce for Energy-Aware and Sustainable Service Delivery for Smart City Applications. *Sustainable Cities and Society*, *77*(November 2021), 103505. https://doi.org/10.1016/j.scs.2021.103505

Resat, H. G. (2020). Design and Analysis of Novel Hybrid Multi-Objective Optimization Approach for Data-Driven Sustainable Delivery Systems. *IEEE Access*, *8*, 90280–90293. https://doi.org/10.1109/ACCESS.2020.2994186

Ribeiro, G. M., & Laporte, G. (2012). An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, *39*(3), 728–735.

Roberti, R., & Ruthmair, M. (2021). Exact methods for the traveling salesman problem with drone. *Transportation Science*, *55*(2), 315–335.

Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, *40*(4), 455–472.

Rossi, F., Van Beek, P., & Walsh, T. (2006). *Handbook of constraint programming*. Elsevier.

Rovira-Sugranes, A., Razi, A., Afghah, F., & Chakareski, J. (2022). A review of AI-enabled routing protocols for UAV networks: Trends, challenges, and future outlook. *Ad Hoc Networks*, *130*(2008784), 102790. https://doi.org/10.1016/j.adhoc.2022.102790

Sacramento, D., Pisinger, D., & Ropke, S. (2019). An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research Part C: Emerging Technologies*, *102*(March), 289–315. https://doi.org/10.1016/j.trc.2019.02.018

Salama, M., & Srinivas, S. (2020). Joint optimization of customer location clustering and drone-based routing for last-mile deliveries. *Transportation Research Part C: Emerging Technologies*, *114*(January), 620–642. https://doi.org/10.1016/j.trc.2020.01.019

Same-day delivery: Ready for a new omnichannel strategy | McKinsey. (n.d.). Retrieved June 26, 2023, from https://www.mckinsey.com/industries/retail/our-insights/same-day-delivery-ready-for-takeoff

Samouh, F., Gluza, V., Djavadian, S., Meshkani, S., & Farooq, B. (2020). Multimodal autonomous last-mile delivery system design and application. *2020 IEEE International Smart Cities Conference (ISC2)*, 1–7.

Sariklis, D., & Powell, S. (2000). *A heuristic method for the open vehicle routing problem*. 564–573.

Savuran, H., & Karakaya, M. (2016). Efficient route planning for an unmanned air vehicle deployed on a moving carrier. *Soft Computing*, *20*(7), 2905–2920.

Schermer, D., Moeini, M., & Wendt, O. (2019). A matheuristic for the vehicle routing problem with drones and its variants. *Transportation Research Part C: Emerging Technologies*, *106*(May), 166–204. https://doi.org/10.1016/j.trc.2019.06.016

Schermer, D., Moeini, M., & Wendt, O. (2020). A branch-and-cut approach and alternative formulations for the traveling salesman problem with drone. *Networks*, *76*(2), 164–186. https://doi.org/10.1002/net.21958

Scherr, Y. O., Hewitt, M., Neumann Saavedra, B. A., & Mattfeld, D. C. (2020). Dynamic discretization discovery for the service network design problem with mixed autonomous fleets. *Transportation Research Part B: Methodological*, *141*, 164–195. https://doi.org/10.1016/j.trb.2020.09.009

Scherr, Y. O., Neumann Saavedra, B. A., Hewitt, M., & Mattfeld, D. C. (2019). Service network design with mixed autonomous fleets. *Transportation Research Part E: Logistics and Transportation Review*, *124*(August 2018), 40–55. https://doi.org/10.1016/j.tre.2019.02.001

Schrage, L. (1981). *Formulation and Structure of More Com plex / Real is t ic Routing and Scheduling Problems*. *11*, 229–232.

Schwerdfeger, S., & Boysen, N. (2020). Optimizing the changing locations of mobile parcel lockers in last-mile distribution. *European Journal of Operational Research*, *285*(3), 1077–1094. https://doi.org/10.1016/j.ejor.2020.02.033

Shavarani, S. M., Nejad, M. G., Rismanchian, F., & Izbirak, G. (2018). Application of hierarchical facility location problem for optimization of a drone delivery system: a case study of Amazon prime air in the city of San Francisco. *The International Journal of Advanced Manufacturing Technology*, *95*, 3141–3153.

Shi, W., Zhou, H., Li, J., Xu, W., Zhang, N., & Shen, X. (2018). Drone assisted vehicular networks: Architecture, challenges and opportunities. *IEEE Network*, *32*(3), 130–137.

Shui, W., & Li, M. (2020). Integrated Pricing and Distribution Planning for Community Group Purchase of Fresh Agricultural Products. *Scientific Programming*, *2020*. https://doi.org/10.1155/2020/8839398

Simoni, M. D., Kutanoglu, E., & Claudel, C. G. (2020). Optimization and analysis of a robot-assisted last mile delivery system. *Transportation Research Part E: Logistics and Transportation Review*, *142*(May 2019), 102049. https://doi.org/10.1016/j.tre.2020.102049

Singgih, I. K., Lee, J., & Kim, B. I. (2020). Node and Edge Drone Surveillance Problem with Consideration of Required Observation Quality and Battery Replacement. *IEEE Access*, *8*, 44125–44139. https://doi.org/10.1109/ACCESS.2020.2976122

Sluijk, N., Florio, A. M., Kinable, J., Dellaert, N., & Van Woensel, T. (2022). Two-echelon vehicle routing problems: A literature review. *European Journal of Operational Research*, *xxxx*. https://doi.org/10.1016/j.ejor.2022.02.022

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

Soto, M., Sevaux, M., Rossi, A., & Reinholz, A. (2017). Multiple neighborhood search, tabu search and ejection chains for the multi-depot open vehicle routing problem. *Computers & Industrial Engineering*, *107*, 211–222.

Tamke, F., & Buscher, U. (2021). A branch-and-cut algorithm for the vehicle routing problem with drones. *Transportation Research Part B: Methodological*, *144*, 174–203. https://doi.org/10.1016/j.trb.2020.11.011

Tarantilis, C. D., Kiranoudis, C. T., & Vassiliadis, V. S. (2002). A backtracking adaptive threshold accepting algorithm for the vehicle routing problem. *Systems Analysis Modelling Simulation*, *42*(5), 631–664.

Tarantilis, C. D., Kiranoudis, C. T., & Vassiliadis, V. S. (2004). A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *European Journal of Operational Research*, *152*(1), 148–158.

Thakur, R. (2018). The role of self-efficacy and customer satisfaction in driving loyalty to the mobile shopping application. *International Journal of Retail & Distribution Management*, *46*(3), 283–303.

Troudi, A., Addouche, S. A., Dellagi, S., & El Mhamedi, A. (2018). Sizing of the drone delivery fleet considering energy autonomy. *Sustainability (Switzerland)*, *10*(9), 1–17. https://doi.org/10.3390/su10093344

Ulmer, M. W., & Streng, S. (2019). Same-Day delivery with pickup stations and autonomous vehicles. *Computers and Operations Research*, *108*, 1–19. https://doi.org/10.1016/j.cor.2019.03.017

Ulmer, M. W., & Thomas, B. W. (2018). Same-day delivery with heterogeneous fleets of drones and vehicles. *Networks*, *72*(4), 475–505.

Valavanis, K. P., & Vachtsevanos, G. J. (2015). *Handbook of unmanned aerial vehicles* (Vol. 1). Springer.

Van Breedam, A. (1995). Improvement heuristics for the vehicle routing problem based on simulated annealing. *European Journal of Operational Research*, *86*(3), 480–490.

Vu, L., Vu, D. M., Minh Hoàng, H., & Nguyen, V. P. (2022). The two-echelon routing problem with truck and drones. *International Transactions in Operational Research*, *29*(5), 2968–2994. https://doi.org/10.1111/itor.13052

Wang, K., Pesch, E., Kress, D., Fridman, I., & Boysen, N. (2022). The Piggyback Transportation Problem: Transporting drones launched from a flying warehouse.

European Journal of Operational Research, *296*(2), 504–519. https://doi.org/10.1016/j.ejor.2021.03.064

Wang, Z., & Sheu, J. B. (2019). Vehicle routing problem with drones. *Transportation Research Part B: Methodological*, *122*, 350–364. https://doi.org/10.1016/j.trb.2019.03.005

Wassan, N., Wassan, N., Nagy, G., & Salhi, S. (2017). The Multiple Trip Vehicle Routing Problem with Backhauls : Formulation and a Two-Level Variable Neighbourhood Search. *Computers and Operation Research*, *78*, 454–467. https://doi.org/10.1016/j.cor.2015.12.017

Wei, L., Zhang, Z., Zhang, D., & Leung, S. C. H. (2018). A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, *265*(3), 843–859.

Welch, A. (2015). *A cost-benefit analysis of Amazon Prime Air*.

Xia, Y., Zeng, W., Xing, X., Zhan, Y., Tan, K. H., & Kumar, A. (2021). Joint optimisation of drone routing and battery wear for sustainable supply chain development: a mixed-integer programming model based on blockchain-enabled fleet sharing. In *Annals of Operations Research*. Springer US. https://doi.org/10.1007/s10479-021-04459-5

Yu, J. J. Q. (2019). Two-stage request scheduling for autonomous vehicle logistic system. *IEEE Transactions on Intelligent Transportation Systems*, *20*(5), 1917–1929. https://doi.org/10.1109/TITS.2018.2849091

Yu, S., Puchinger, J., & Sun, S. (2020). Two-echelon urban deliveries using autonomous vehicles. *Transportation Research Part E: Logistics and Transportation Review*, *141*(July 2019), 102018. https://doi.org/10.1016/j.tre.2020.102018

Yu, S., Puchinger, J., & Sun, S. (2021). Van-based robot hybrid pickup and delivery routing problem. *European Journal of Operational Research*.

Yu, S., Puchinger, J., & Sun, S. (2022). Van-based robot hybrid pickup and delivery routing problem. *European Journal of Operational Research*, *298*(3), 894–914. https://doi.org/10.1016/j.ejor.2021.06.009

Zhang, G., Zhu, N., Ma, S., & Xia, J. (2021). Humanitarian relief network assessment using collaborative truck-and-drone system. *Transportation Research Part E: Logistics and Transportation Review*, *152*(October 2020), 102417. https://doi.org/10.1016/j.tre.2021.102417

Zhang, J., Onal, S., Das, R., Helminsky, A., & Das, S. (2019). Fulfilment time performance of online retailers–an empirical analysis. *International Journal of Retail & Distribution Management*, *47*(5), 493–510.

Zhang, M., & Yang, B. (2022). Swarm Robots Cooperative and Persistent Distribution Modeling and Optimization Based on the Smart Community Logistics Service Framework. *Algorithms*, *15*(2). https://doi.org/10.3390/a15020039

Zhen, L., Ma, C., Wang, K., Xiao, L., & Zhang, W. (2020). Multi-depot multi-trip vehicle routing problem with time windows and release dates. *Transportation Research Part E*, *135*(January), 101866. https://doi.org/10.1016/j.tre.2020.101866

Zhou, L., Baldacci, R., Vigo, D., & Wang, X. (2018). A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution. *European Journal of Operational Research*, *265*(2), 765–778.