Stochastic optimisation of unmanned aerial vehicle assisted package delivery planning

Sawadsitang, Suttinee

2021

Sawadsitang, S. (2021). Stochastic optimisation of unmanned aerial vehicle assisted package delivery planning. Doctoral thesis, Nanyang Technological University, Singapore.
<https://hdl.handle.net/10356/146991>

<https://hdl.handle.net/10356/146991>

<https://doi.org/10.32657/10356/146991>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 22:51:09 SGT

Stochastic Optimisation of Unmanned Aerial Vehicle Assisted Package Delivery Planning

Suttinee Sawadsitang

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

14/09/2020

.....

Date

.....

Suttinee Sawadsitang

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

14/09/2020

.....

Date

.....

Prof. Dusit Niyato

Authorship Attribution Statement

This thesis contains material from the three papers, i.e., (i) published in a peer-reviewed journal, (ii) under reviewed in a peer-reviewed journal, and (iii) accepted at a conference, which I am listed as the main author.

Chapter 3 is published as [S. Sawadsitang, D. Niyato, P.S. Tan, and P. Wang. "Joint ground and aerial package delivery services: A stochastic optimization approach." IEEE Transactions on Intelligent Transportation Systems 20.6 \(2018\): 2241-2254.](#)

The contributions of the co-authors are as follows:

- Prof. Niyato gave the initial project direction and edited the manuscript drafts.
- Dr. Tan provided the practical data, which is collected from a Singapore logistics company. She also provided guidance on the paper direction and reviewed the final manuscript.
- Assoc. Prof Wang helped design the organisation of the paper and edited the final manuscript.

Chapter 4 is Proceedings as [S. Sawadsitang, D. Niyato, P.S. Tan, P. Wang, and S. Nutanong, "Multi-Objective Optimization for Drone Delivery." 2019 IEEE 90th Vehicular Technology Conference \(VTC2019-Fall\) \(pp. 1-5\). IEEE.](#)

The contributions of the co-authors are as follows:

- Prof. Niyato suggested the project direction and edited the manuscript drafts.
- Dr. Tan provided the practical data, which is collected from a Singapore logistics company. She also provided guidance on the paper direction and reviewed the final manuscript.
- Assoc. Prof. Wang and Assoc. Prof. Nutanong helped design the organisation of the paper and edited the final manuscript.

Chapter 5 is under reviewed as [S. Sawadsitang, D. Niyato, P.S. Tan, P. Wang, and S. Nutanong, "Shipper Cooperation in Stochastic Drone Delivery: A Dynamic Bayesian Game Approach" IEEE Transactions on Vehicular Technology](#)

The contributions of the co-authors are as follows:

- Prof. Niyato suggested the project direction and edited the manuscript drafts.

- Dr. Tan provided the practical data, which is collected from a Singapore logistics company. She also provided guidance on the paper direction and reviewed the final manuscript.
- Assoc. Prof. Wang and Assoc. Prof. Nutanong edited the final manuscript and gave significant and valuable comments.

14/09/2020

.....

Date

.....

Suttinee Sawadsitang

Acknowledgements

I would like to express my sincere gratitude to my supervisor professor Dusit Niyato for his constant support and guidance throughout my Ph.D. study at Nanyang Technological University (NTU). I would like to thank him for his immense enthusiasm and patience. He always motivates me even when I am lost. I could not have imagined having a better adviser and mentor for my Ph.D. study.

Moreover, I would like to give a sincere thank to my co-supervisors, Dr. Puay Siew Tan and associate professor Ping Wang for their constant guidance as well as for providing the necessary information and the direction.

Beside my supervisor and my co-supervisors, I would like to thank all the collaborators and my lab-mates, who have shared research idea and knowledge with me. Especially, I would like to thank associate professor Sarana Nutanong for giving valuable comments on my research works. In addition, I am deeply indebted to Mr. Rakpong Kaewpuang and Dr. Thanomwong Poorisat who helped me settle down when I first moved to Singapore. They gave me practical and helpful guidance about research and how to live in this multi-culture country.

Abstract

The need for automated delivery services has never been greater. As the COVID-19 pandemic grips the world, manufacturers and retailers are faced with challenges to continue their businesses with social distancing in place. E-commerce, which has already been experiencing unprecedented growth, is now being adopted even more widely. Given an increase demand in delivery services and the need for social distancing measures, drone delivery becomes an attractive option. Unmanned aerial vehicle or drone delivery involves much less human interaction, incurs a much lower cost, offers a faster delivery, and is more eco-friendly than truck delivery. These advantages have led many big shippers such as UPS, Amazon, Japan Post, SingPost, Alibaba, and JD.com to set their eyes on incorporating this novel delivery method to their businesses.

One of the greatest challenges in a race to achieve drone delivery is how to manage the high probability of service interruptions to ensure punctual and reliable delivery. Beside technical issues, drone delivery services may be disrupted and fail due to a wide range of factors such as strong wind, rain, human abuse, and animal attacks. These unpredicted factors introduce uncertainties, which have not been considered in existing research on planning of drone delivery services. The first shipper to consider these uncertainties and derive an effective and efficient delivery plan to address them can gain significant competitive advantages and lead the transformation of delivery logistics services.

In this thesis, I introduce (i) the joint ground and aerial delivery service optimisation and planning (GADOP) framework, (ii) multi-objective optimisation for drone delivery (MODD) framework, and (iii) Bayesian Shipper Cooperation in Stochastic Drone Delivery (BCoSDD) framework, which all of them explicitly incorporate uncertainty of drone package delivery, i.e., take-off and breakdown conditions. Firstly, the GADOP framework aims to minimise the total delivery cost given practical constraints, e.g., traveling distance limit. Specifically, I formulate the GADOP framework as a three-stage stochastic integer programming model. To deal with

the high complexity issue of the problem, a decomposition method is adopted. To extend the study in stochastic drone delivery, I propose the MODD framework as a joint aerial delivery and outsourcing delivery as a three-objective optimisation. I apply the E-constraint method to handle the multi-objective optimisation in the MODD framework. I also consider a scenario in which multiple shippers can cooperate to minimise their drone delivery cost. The BCoSDD framework is composed of three functions, i.e., package assignment, shipper cooperation formation and cost management. The uncertainties of misbehaviour of cooperative shippers are also taken into account by dynamic Bayesian coalition formation game.

The performance of the frameworks is evaluated using two data sets including Solomon benchmark suite and the real data from a Singapore logistics company. The tradeoffs among drone delivery, conventional ground-based delivery, and outsourcing carrier are presented for all proposed frameworks. Simulation results have shown that the GADOP framework always achieves a lower total payment than that of the expected value formulation (EVF) and the Parallel Drone Scheduling Traveling Salesman Problem (PDSTSP). The computational time gaps between the decomposed GADOP and the original GADOP are grown exponentially when the numbers of drones and uncertainty scenarios increase. Our experiments show that the decomposed GADOP solve about a hundred times faster than the original GADOP when the numbers of drones and scenarios are larger than 6 and 64, respectively. For the BCoSDD framework, the experiment results show that it can achieve the most effective solution, i.e., the lowest delivery cost, compared with other baseline approaches, i.e., deterministic drone delivery (DDD), stochastic drone delivery (SDD), shipper cooperation in deterministic drone delivery (CoDDD), and shipper cooperation in stochastic drone delivery (CoSDD). To sum up, the studies can help single/multiple shipper(s) plan the drone delivery more effectively.

Contents

Acknowledgements	ix
Abstract	xi
List of Figures	xvii
List of Tables	xix
Symbols and Acronyms	xxi
1 Introduction	1
1.1 Background of delivery modes	2
1.1.1 Last-mile deliveries	2
1.1.2 Delivery business models	3
1.2 Motivation, objectives, and scopes	4
1.2.1 Objectives	7
1.2.2 Scopes	8
1.3 Contributions	8
1.4 Organisation of the thesis	11
2 Literature Review	13
2.1 Overview of literature review	13
2.2 Types of vehicle routing problem	17
2.2.1 Traditional routing problem and its formulations	17
2.2.2 Variety of the existing vehicle routing problem	18
2.3 Detail of aerial-based routing and delivery	24
2.3.1 Drone routing	24
2.3.2 Drone delivery	25
2.4 The modelling characteristics of optimisation parameters and objective function	26
2.4.1 Uncertainty in optimisation parameters	27
2.4.1.1 Stochastic VRP	28
2.4.1.2 Dynamic VRP	29
2.4.1.3 Stochastic and dynamic VRP	31

2.4.2	Multiple objectives for VRP	32
2.5	Modelling delivery planing for multiple shippers	34
2.5.1	Game theory approach	34
2.5.2	Auction approach	35
2.6	Solution techniques	36
2.7	The summarise of the research gaps	40
2.7.1	Stochastic optimisation for a single shipper	40
2.7.2	Cooperative resource planning for multiple shippers	43
3	Single Objective Optimisation for Truck and Drone Delivery ¹	45
3.1	System model and assumptions	46
3.1.1	Uncertainty	49
3.1.2	Payments	51
3.2	GADOP problem formulation: single objective	51
3.3	Solution technique: decomposition	57
3.3.1	Master problem or first-stage problem	59
3.3.2	Second-stage sub-problem	60
3.3.3	Third-stage sub-problem	61
3.3.4	Calculating feedback parameters and convergence	62
3.3.5	The complexity of the decomposition	64
3.4	Evaluation	65
3.4.1	Parameter setting	65
3.4.2	Tradeoff between taking and not-taking uncertainty into account	66
3.4.3	Routing, payment, and explanation of the real dataset	68
3.4.4	Impacts of the number of customers and drone flying limitations	70
3.4.5	Impact of random takeoff scenarios	70
3.4.6	Impact of random breakdown scenarios	74
3.4.7	Impact of penalty cost	74
3.4.8	Impact of repair cost	76
3.4.9	Simulation	76
3.4.10	Decomposition	78
3.5	Conclusion	78
4	Multi-objective Optimisation for Drone and Outsource Delivery ²	81
4.1	System model and assumptions	82
4.2	Multi-objective optimisation formulations	84
4.2.1	Multiple objective functions	85
4.2.2	Decision variables	86
4.2.3	Constraints	87

¹The work in this chapter has been presented in [123]

²The work in this chapter has been presented in [124]

4.2.4	Solution Technique: ϵ -constraint method	89
4.3	Performance evaluation	90
4.3.1	Pareto frontier	91
4.3.2	Reward of on-time delivery	92
4.3.3	Impact of probabilities (λ and ω)	94
4.4	Conclusion	95
5	Drone and Outsource Delivery Planning for Multiple Shippers ³	97
5.1	System model and assumptions	99
5.2	Package assignment	101
5.2.1	Package assignment background	101
5.2.2	Package delivery operation and scenario tree	102
5.2.3	Package assignment formulations	103
5.2.4	Simplification	107
5.2.4.1	The proof of the transformation	109
5.3	Cost management	112
5.4	Shipper Cooperation	113
5.4.1	Cooperation background (Coalition game)	114
5.4.2	Expected payoff (μ_p)	115
5.4.3	Preference relations	116
5.4.4	Solution technique: merge and split algorithm	117
5.4.5	Solution technique: markov transition process	119
5.5	Belief update mechanism	120
5.6	Performance evaluation	123
5.6.1	Experiment setup	123
5.6.2	Analysis of cooperation management	125
5.6.3	Impact of the optimisation	128
5.6.4	Simulation	129
5.6.5	Belief update mechanism	135
5.7	Conclusion	136
6	Conclusion and Future Works	139
6.1	Conclusion	139
6.2	Future works	141
A	Other Optimisation Techniques	145
A.1	Expected value formulation (EVF)	145
A.2	Parallel drone scheduling traveling salesman problem (PDSTSP)	146
	List of Author's Publications	147
	Bibliography	149

³The work in this chapter has been presented in [125]

List of Figures

1.1	Overall diagram of how thesis is organised.	6
2.1	Mind map of the literature review	14
2.2	Classification of pickup and delivery problem [20].	20
2.3	Solution quality and computational time performance of metaheuristic methods [136].	38
3.1	The joint ground and aerial delivery service optimisation and planning (GADOP) framework	47
3.2	The time-line of the scenarios.	47
3.3	An example of scenario tree with one drone.	47
3.4	Decomposition of scenarios and stages where a circle represents one optimisation master problem or sub-problem.	58
3.5	Solomon benchmark example with the scenario that the drones are always able to takeoff and no breakdown occurs.	67
3.6	Solomon benchmark example with two takeoff and breakdown scenarios.	69
3.7	The real package delivery problem in Singapore. (a) and (b) present the customers who are served by the drone and the truck, respectively.	71
3.8	Increasing number of customers.	72
3.9	Performance comparison between truck delivery and the proposed GADOP framework.	73
3.10	Increasing the probability of that all drones cannot take off.	73
3.11	The results of varying the number of breakdown customers and the probability of the breakdown scenarios, where the bars with an asterisk symbol on the top represents the results that all customers are served by the truck.	74
3.12	Effects of increasing the penalty.	75
3.13	The drone repair cost is varied.	75
3.14	The comparison between the solutions from the GADOP, the EVF, and the PDSTSP.	76
3.15	The comparison between the original GADOP and the decomposition. The number of takeoff scenarios and the number of drones are varied.	77
4.1	Map	91

4.2	The Pareto frontier	92
4.3	Impact of number of matching required time windows while minimising (a) the total cost and (b) the percentage of unsuccessful delivered packages.	93
4.4	The impact of breakdown probability on the percentage of unsuccessful delivered packages.	94
5.1	The shipper cooperation in static BCoSDD framework.	100
5.2	An example of a scenario tree of a drone.	103
5.3	The state transition diagram of three shippers.	117
5.4	Tree diagram of the belief update mechanism of shipper p about shipper q	121
5.5	The locations of customers and depots.	124
5.6	Analysis of the stable coalition formation	127
5.7	Impact of Optimisation	130
5.8	Simulation results of deterministic (DDD) and stochastic (SDD) approaches of shipper P_3 with one drone.	132
5.9	The comparison of simulation results when the initial cost of drone = S\$100	133
5.10	The comparison of simulation results when shippers own the drones (the initial cost of drone = S\$0)	133
5.11	Belief update mechanism, where shippers p_1 and p_2 always deliver packages, shippers p_3 and p_4 deliver only 75% of packages that belong to the other cooperative shippers.	136

List of Tables

2.1	Classification of problems with uncertainty	27
2.2	The comparison of the related work and my study	33
2.3	References and Acronyms of metaheuristics, which are indicated Figure 2.3	39
2.4	The comparison between existing studies and the proposed frameworks when a single shipper is considered	41
2.5	The comparison between existing studies and the proposed frameworks when multiple shippers are considered	43
4.1	The notations in the MODD framework	85
5.1	Truth table to the joint probabilities of the belief update mechanism.	122
5.2	Shapley value and coalition structures	127
5.3	Shapley value of the different approaches for the decisions in the simulation experiment (when the initial cost of drones is set as S\$100))	134
5.4	Shapley value of the different approaches for the decisions in the simulation experiment (when the shippers own drones, i.e., the initial cost is set to S\$0))	134

Symbols and Acronyms

Acronyms

GADOP	Joint ground and aerial delivery service optimisation and planning
MODD	Multi-objective optimisation for drone delivery
BCoSDD	Bayesian shipper cooperation in stochastic drone delivery
EVF	Expected value formulation
PDSTSP	Parallel drone scheduling traveling salesman problem
DDD	Deterministic drone delivery
SDD	Stochastic drone delivery
CoSDD	Shipper cooperation in stochastic drone delivery
UAV	Unmanned aerial vehicle
CAGR	Compound annual growth rate
VRP	Vehicle routing problem
VRPPC	Vehicle routing problem with private fleet and common carrier
VRPTW	Vehicle routing problem with time windows
MDVRP	Multi-depots vehicle routing problem
LRP	Location routing problem
SRI	Selection, routing and improvement
IDEA	Iterated density estimation evolutionary algorithm
FSTSP	Flying sidekick traveling salesman problem
SA	Simulated annealing
CCP	Chance constrained program
MCDM	Multi-criteria decision making
MOSPOOLS	Multi-objective smart pool search
TSP	Traveling sale man problem
SRRA	Single request reassignment
BRRA	Bundle request reassignment

ILP	Integer linear programming
EMA	Exponential moving average
SAA	Sample-average approximation

Chapter 1

Introduction

Package transportation is a fast-growing business. In 2018, package delivery market was valued at US\$ 31.2 billion, and it is predicted to reach US\$ 61.5 billion by 2025 [41]. One of the key factors driving its growth is the increase in popularity of e-commerce worldwide. The efficiency of effective logistics delivery makes it possible for business to improve customer satisfaction. By 2024, the global e-commerce logistics market is predicted to expand to US\$781 billion or more than six times its value in 2014, which was US\$ 122.2 billion [134]. In the case of food and beverages industry, the cost of delivery and distribution added to the final price can range up to 70% of the actual product cost [5]. The rise in cost incurred from package delivery has the potential to hurt sales because customers typically make their purchase decision based on the total cost, which includes delivery. This highlights the need for research and applications to advance logistics in package delivery. If business owners and shippers have tools to help them better plan their delivery, they could cut cost and achieve a higher profit margin [57]. Moving from manual planning to an advance optimisation tool, the shipper can reduce transportation cost by 10 to 30 percentages [105].

Land transportation using trucks, cars, motorcycles, and bicycles is a conventional method of package delivery. Some businesses have considered using drones or unmanned aerial vehicles (UAVs). UAVs are aerial vehicles that can fly autonomously or be piloted remotely. Many fields such as surveillance, entertainment, wireless communication, and package delivery have all employed UAVs. Business Insider Intelligence has predicted that the sale of drones will reach US\$12 billion by 2021,

which is up by a compound annual growth rate (CAGR) of 7.6% [9]. Thanks to the advancement of current technologies, the efficiency and reliability of using drones in package delivery has improved significantly. Big businesses such as Amazon Prime Air, DHL Parcelcopter, Alibaba, Japan Post, and Google Project Wing have all used drones as part of their delivery process [118][132]. This strongly suggests the technological and economic feasibility of using drone for package delivery. According to the unmanned airspace survey reported in 2019 [139], 45 countries are either implementing or preparing to use drones to deliver parcels. Currently, drone delivery is used the most in a medical field.

The benefits of drone parcel delivery are listed as follows.

First, drones are expected to deliver faster than land vehicles because they can avoid traffic jam and deliver a package directly to a customer drop-off station with minimal human interaction [146].

Secondly, drones delivery use less labor and energy compared to ground-based delivery. Thus, it is more environment-friendly [63].

Thirdly, drones delivery can access certain areas where land vehicles cannot.

This chapter first presents the background information with regards to delivery modes. After that, the motivation, objectives, and scopes of the studies are presented in Section 1.2. Section 1.3 presents the contribution of the studies. The organisation of the thesis is presented at the end of this chapter.

1.1 Background of delivery modes

1.1.1 Last-mile deliveries

Last-mile delivery represents the last step of the logistics process to distribute packages directly to the end-customers. Even the name is implied as the last mile of a delivery, last mile logistics or last mile delivery can range from a few blocks to a hundred mile [4]. Last mile delivery usually involves three stakeholders, i.e., shippers, carriers, and the end-customers. Shipper refers to the person or company who has a package and want the package to be delivered to the customer. Carrier

refers to a person or company who carries the package of the shipper to the end-customer. A shipper also can act as a carrier. End-customer is the person who waiting for the package to arrive at the door step.

Due to the dramatic growth of e-commerce, the demand on the last mile logistics is increasing accordingly. Especially, COVID-19 pandemic has recently created many new normal to us. Social distancing, which is one of the strategies to prevent the spread of a contagious disease, significantly increases the demand on last mile logistics. People order foods and essentials online and expect those items to be delivered reliably and on time.

Since the demand of the last mile delivery has been growing tremendously, the capacity constraint and driver shortage become significant challenges. To overcome these challenges, an optimisation tool can help the shipper to better plan the delivery allocations and delivery routes. Drones and robots are also new alternative solution in last mile delivery.

1.1.2 Delivery business models

Shippers can choose to deliver packages to customers with different delivery business models as listed below.

- Self-managed ground delivery
- Self-managed drone delivery
- Outsourced ground delivery
- Outsourced drone delivery

Self-managed ground delivery can be referred to as truck delivery. As the name represents, shippers need to manage the delivery by themselves, which are to (i) reserve or prepare delivery vehicles, i.e., trucks, and (ii) determine the routes of the trucks. In general, the routes are determined before the trucks depart from a depot. The shippers are required to pay for (i) the initial cost, e.g. rental fee and driver wages, and (ii) routing cost, e.g. fuel and road express fee. For each route, a truck travels from a depot to a number of geographically scattered customers and

returns back to the original depot at the end. Truck delivery is likely to be more reliable and accessible than drone delivery [32].

Self-managed drone delivery mode has become an alternative delivery for shippers. The shipper is required to reserve or prepare drones in advance. If the delivery is scheduled for tomorrow, the shipper will have to rent the drone or charge the batteries at least the night before. Each drone will depart from a shipper's depot to a customer location. After that, the drone will drop the customer package and return to the depot. To deliver packages by a drone, the shipper may need to pay for (i) an initial cost, e.g., rental fee, (ii) operational cost, e.g., electricity for drone's battery, and (iii) maintenance fee and penalty in cases of drone breakdown and failure to deliver. In fact, drones are less reliable when comparing to the other delivery modes as drones are more susceptible to weather conditions. However, drone delivery is promising for a reduction in manpower.

Another delivery mode for shippers is to outsource packages to a third-party carrier in an on-demand basis. The pricing can be calculated based on weight or volume of a package. In this thesis, the studies do not focus on the vehicle-based delivery of the third-party carrier, i.e., a truck or a drone. Instead, the studies only focus on the delivery fee charged by the third-party carrier.

1.2 Motivation, objectives, and scopes

More than 50 years after the release of vehicle routing problem (VRP), many surveys reported the variety of the actual necessity in VRP and its complexity due to multiple specific constraints are required. Various kinds of VRP (i.e., Capacity VRP, Time-dependent VRP, Multi-depot VRP, Periodic VRP, Multi-echelon VRP) have been identified and explored. Drone delivery is a kind of VRP as the shipper's objective is to minimise the total delivery cost or the total flying distance and obtain the flying route of drones. Due to the specific requirements of drone delivery, formulating a system model and creating a framework are the challenges for us to investigate and improve.

Despite their advantages over ground-based delivery, drones face many challenges and suffer from some constraints. Compared to ground-based delivery and the outsourcing mode, drone delivery is less reliable and has significantly less service

coverage and lower capacity. The flying distance limits and the location of a shipper's depot are therefore the important parameters affecting a serving area of the drone delivery. If customers are located outside the serving area, the shippers cannot use a drone to deliver the package. Drone delivery also has a much lower payload threshold when comparing to the other delivery modes.

To enhance drone delivery, the shipper can address the aforementioned challenges as follows:

- (1) **Hybrid drone and ground-based delivery:** integrating drone delivery with the ground-based delivery was introduced in [11]. These two delivery modes can complement each other because of their unique features and advantages. Packages can be optimally assigned to be delivered by trucks or drones to maximise service quality and to minimise cost. Thus, a joint ground and aerial delivery service optimisation and planning emerge as an important area.
- (2) **Outsourcing inclusion option:** the integration of the outsourcing mode and drone delivery mode is also an alternative solution for the shipper. Outsourcing mode may be used when the customers are unreachable or uneconomical by drone delivery. The integration of traditional ground-based delivery mode and outsourcing mode was introduced in [95]. This was known as vehicle routing problem with private fleet and common carrier (VRPPC). The integration of drone delivery mode and outsourcing mode was introduced in our work [122]. The evaluation shows that the combination works effectively as outsourcing mode serves the customers which drone delivery mode cannot.
- (3) **Cooperative collaboration to improve the performance of (1) and (2):** expanding the coverage area by considering more depots in the system is a straightforward solution. Having multiple depots in different areas allow us to use drones to serve scattered locations. When the shipper has more than one depot, a system that considers more number of depots will give us a better drone delivery planning. However, to launch new depots is not a simple and practical solution in industries because it is a long-term investment and requires investment. To avoid such complicated depot-building processes and a significant investment in depot-building, the shipper can cooperate with

FIGURE 1.1: Overall diagram of how thesis is organised.

other shippers to create a pool of shared resources, i.e., depots, drones, and customers. By cooperating, the shippers can to achieve a lower delivery cost.

Different shippers may have different environments, different missions, and different needs in their businesses. For example, some shippers may focus only to minimise the total delivery cost while some shippers may want to maintain high customer satisfaction in the long term, i.e., maximising the reward of on-time delivery. Some shippers may have an opportunity to cooperate with their business partners to achieve a better delivery planning. As a result, to help general shippers in the market build a tool to plan their delivery, different assumptions are needed for different shippers. Therefore, in this thesis, I fulfil the research area of drone delivery by investigating and proposing three frameworks, which are presented in Chapter 3 to Chapter 5. The proposed frameworks have different assumptions, different setting, and different methodologies, which aim to help diverse shippers from different business backgrounds. Each of the frameworks will be suitable for different scenarios and they cannot be used/applied interchangeably.

The difference in term of applications between the chapters are illustrated as a diagram in Figure 1.1. The detail is discussed as follows.

In Chapter 3, I consider a scenario that the shipper has only one depot, a set of trucks, and a set of drones. Thus, a joint truck and drone delivery framework

for the shipper to serve its customers is proposed. Each customer will be served by either one truck or one drone. If the assigned drone fails to deliver to an assigned customer, the shipper needs to pay the penalty.

In Chapter 4, I consider the scenario that a shipper owns more than one depot, and drones can depart from any of them. In this scenario, I propose a framework with joint drone delivery and outsourcing service since drones may not be able to serve every customer. Similar to chapter 3, if the assigned drone fails, the shipper needs to pay the penalty.

In Chapter 5, I extend the application in Chapter 4 by allowing shippers to cooperate, and thus the serving area of drones is expanded. Multiple shippers may join to create a resource pool, i.e., shipper depots, drones, and customers.

1.2.1 Objectives

The main goal of the studies is to help a shipper achieve optimally package delivery plan in last mile logistics by the following objectives.

- Objective 1 : To improve and to extend a hybrid solution between different delivery modes for the shipper to perform the package delivery since nowadays multiple delivery modes are available.
- Objective 2 : To optimise the drone delivery problems with uncertainty factors, i.e., breakdown and departing condition, which has not been considered in the existing works.
- Objective 3 : To reduce computational time of the package delivery management problem as once uncertainties are considered, the computational times are too slow and likely to be unacceptable in industries.
- Objective 4 : To manage the package delivery when the shipper wants to consider more than one cost factor.
- Objective 5 : To introduce a new methodology and a new algorithm to manage package delivery among shipper cooperation.

Objective 6 : To propose a new model to learn the shipper behaviour when the shipper agrees to join the package delivery cooperation. The belief update machine is needed since it is impossible to guarantee that the shipper will not misbehave, i.e., does not deliver a package that belongs to the other shipper.

1.2.2 Scopes

The scopes of this thesis are listed as follows:

- All the proposed frameworks are capable of multiple drones and multiple customers.
- This thesis focuses on the optimisation modelling aspect. The objective function and constraints are presented in mathematical formulations with explanations. The given inputs, i.e., sets and parameters, are also denoted in the mathematical forms.
- The commercial mathematical optimisation solver, i.e., integer programming, which is named as CPLEX, is used to perform the experiments. The optimisation solver can provide an optimal solution without implementation effort if users have mathematical formulations.
- This thesis does not focus and experiment on heuristic solution methods. A heuristic method is likely to execute faster than the integer programming. However, heuristic methods are likely to require more human effort when the user want to add a new feature, i.e., constraint, after the framework has been deployed.
- The comparison between baseline methods and the proposed methods are presented.

1.3 Contributions

The contributions of this thesis are presented as follows:

- In Chapter 3, The joint ground and aerial delivery service optimisation and planning (GADOP) framework for a shipper is proposed. The GADOP framework considers two delivery modes, i.e., truck and drone deliveries.
 - The proposed joint optimisation is able to utilise and trade off benefits and limitations of truck and drone delivery. In particular, the proposed optimisation can yield the minimum total payment for the shipper by considering the routing path of trucks and the serving order of drones.
 - The formulated stochastic programming model can find and achieve an optimal solution by taking the uncertainty of the drone delivery into account. Consequently, the shipper can obtain the better planning for the truck and drone usage given certain constraints such as capacity limit and traveling distance limit.
 - To reduce the computational time of obtaining the solution of the stochastic programming model, the decomposed GADOP is introduced by adopting the L-shape method. The decomposition allows the model to be split into multiple smaller sub-problems, which are more computationally tractable.
- In Chapter 4, A multi-objective optimisation for drone delivery (MODD) framework is proposed. The MODD framework aims to help the shipper schedule and plan its parcel delivery. •
 - The optimisation is formulated as a three-stage stochastic programming to handle the uncertainties of the problematic events, i.e., takeoff condition and breakdown condition.
 - The trade-off between using drones and outsourcing packages to a carrier is analysed in the framework as drones may not be able to reach some customers, e.g., due to flying distance limit. Moreover, the evaluation of the MODD framework is presented with the real data from a Singapore company.
 - Since some industries may consider delivery problem with multiple objectives, the MODD framework is formulated as a multi-objective optimisation. The details of the problem objectives are as follows:
 - * To minimise the total delivery cost.

- * To minimise the percentage of unsuccessful package delivery. The unsuccessful delivery occurs when the drone is unable to take off from the depot or the drone breaks down during the delivery.
- * To maximize the reward of on-time delivery. Customers can have one or more specific preferred time slots for delivering and receiving their packages. The time slots are then associated with different rewards to be optimised by the shipper.

The ε -constraint method is used to obtain the exact solutions, and the analysis of the Pareto frontiers of the MODD framework is presented.

- In Chapter 5, the Bayesian Shipper Cooperation in Stochastic Drone Delivery (BCoSDD) framework is introduced for helping shippers to strategize their delivery planning and resource sharing. Two delivery modes are considered in this framework, namely, drone delivery mode and outsourcing mode.
 - In the proposed BCoSDD framework, I introduce the package assignment based on multistage stochastic programming optimisation. The package assignment takes the uncertainty of the drone breakdown into account to assign packages to be delivered by drones with the lowest cost while meeting all customers' demand. While the original problem of the package assignment is non-linear, I reformulate the problem into a linear one that can be solved more efficiently.
 - A shipper cooperation management is proposed. Specifically, a cooperation decision making process of the shippers is modelled as a static and a dynamic Bayesian coalition formation game, respectively. The static Bayesian coalition formation game is applied when the belief about the shippers is known a priori. Alternatively, the dynamic Bayesian coalition formation game is applied when the shippers have to observe and learn behaviours of the other shippers without complete belief. I analyse the stability of the shipper cooperation and also present an algorithm to reach the stable solution. Moreover, in reality, some shippers may misbehave in the cooperation, e.g., a shipper may not deliver a package assigned, the algorithm is thus designed to learn the shippers' behaviour and allow honest shippers to form the cooperation effectively.

- Among the shippers cooperating with each other, the incurred cost will be distributed. I therefore propose cost management to allow the cooperative shippers to share the cost of package delivery. The Shapley value technique is adopted to reach a fair cost sharing.

1.4 Organisation of the thesis

The thesis is organised as follows.

Chapter 1 provides the overview of drone delivery, the background of different delivery modes, research challenges and motivations, and the major contributions of this thesis.

Chapter 2 presents the literature review based on three categories, i.e., (i) applications, (ii) modeling and the explanation, and (iii) solving algorithms.

Chapter 3 introduces the joint ground and aerial delivery service optimisation and planning (GADOP) framework. The framework explicitly incorporates uncertainty of drone package delivery, i.e., takeoff and breakdown conditions. The GADOP framework aims to minimise the total delivery cost, given practical constraints, e.g., traveling distance limit. The GADOP framework is formulated as a three-stage stochastic integer programming model and adopted a decomposition method to handle the high complexity of the problem.

Chapter 4 presents multi-objective optimisation in drone delivery (MODD) framework for a single shipper to optimally plan their delivery with two delivery modes available, i.e., drone delivery and outsourcing. The objectives are to (i) to minimise the total delivery cost, (ii) to minimise the percentage of unsuccessful delivered packages, and (iii) to maximise the reward of on-time delivery. The MODD framework is formulated as a three-stage stochastic mix-integer programming problem.

Chapter 5 is extended from Chapter 4 by using only one objective function, which is to minimise the total delivery cost, to make it less complicated. I also want to concentrate on shipper cooperation. In this chapter, I consider a scenario that multiple shippers can cooperate to minimise their drone delivery cost. The Bayesian

Shipper Cooperation in Stochastic Drone Delivery (BCoSDD) framework is proposed. The framework is composed of three functions, i.e., package assignment, shipper cooperation formation and cost management.

Chapter 6 concludes the thesis and presents potential future research directions.

Chapter 2

Literature Review

In academic research, routing problems are known as vehicle routing problem (VRP), which vehicles travel from an origin, i.e., depot, and visit all customers only once and return to the origin. VRP was first introduced in [35]. Since then, VRP has enjoyed close attention from many researchers. Hundreds of models and algorithms have been proposed to solve the diverse classes of VRP. Many surveys have been published in different periods of time (i.e., 1998[94], 1992[39], 1996[89], 2000[38], 2005[103][104], 2007[151][34], 2008[7], 2009[6], 2012[120], 2013[142][136], and 2014[20]) to present the cutting-edge investigation of VRP.

Drone delivery problem can be classified as one class of VRP. In general, drone delivery problem aims to help a shipper minimize the total delivery cost. Drones may depart from a depot to serve multiple customers and return to the depot, or depart from a depot to serve only one customer and return to depot to re-load a package of the next customer, and fly off to serve the next customer again. Note that the latter is common at the moment.

2.1 Overview of literature review

In this chapter, the review is divided into three different areas, i.e., problem types, modelling, and solution techniques. The literature review of the thesis can be summarised as a mind map, which is illustrated in Figure 2.1.

FIGURE 2.1: Mind map of the literature review

Problem type component is detailed in Section 2.2 and Section 2.3.

- Section 2.2 presents the general idea of VRP. The traditional vehicle routing problem and the optimisation formulation are reviewed in Section 2.2.1. As VRP has a long and successful history, there are many extensions, I only review some classes of the relevant VRP in Section 2.2.2. According to Figure 2.1, four delivery modes of heterogeneous VRP are presented as follows.
 - Ground-based delivery mode, which can be referred to as truck delivery.
 - Aerial-based delivery mode, which can be referred to as UAV or drone delivery.
 - Sea-based delivery mode.
 - Outsourcing delivery mode, which can be referred to as third-party carrier delivery.
- Section 2.3 presents the review of drone routine and drone delivery more in detail since this thesis focuses on drone delivery.

Modelling component is detailed Section 2.4 and Section 2.5.

- Section 2.4 presents the modelling of optimisation problems for a single shipper. In general, an optimisation has four components as follows:
 1. **Input parameter:** an optimisation can be classified as a stochastic or dynamic problem based on the characteristic of uncertainty. More detail can be found in Section 2.4.1.
 2. **Decision variable:** an optimisation can be classified as discrete or continuous based on the values that are allowed for decision variables. If an optimisation considers discrete variables, it is usually called as an integer or a mix-integer programming. In general, all logistics optimisation problems, i.e., VRP, consider discrete variables.
 3. **Objective function:** an optimisation can be classified as multi-objective optimisation when more than one objective functions are considered in the model. The literature based on multi-objective can be found in Section 2.4.2.

4. **Constraint:** an optimisation can be classified as a constrained and non-constrained problem based on the existence of constraints. However, all VRP and related problems are constrained optimisation.

Moreover, an optimisation can be classified as linear or non-linear based on the formulations in both objective functions and constraints. Researchers usually call optimisation models based on the classification above. For example, stochastic linear programming is referred to the model that is considered stochastic parameters and has only linear formulations in both objective function and all constraints.

Since VRP is always modelled with constraints and discrete variables, then only the literature review of input parameters and objective function are presented in Section 2.4.1 and Section 2.4.2, respectively.

- Section 2.5 presents combination modelling techniques for multiple shippers

Solution technique component are detailed Section 2.6.

Package delivery problems are usually an NP-hard problem. The computational time for solving the problem depends on the numbers of variables, parameters, and constraints. Thus, these problems may need different approximation methods to reduce the computational time. The algorithms of the problems can be classified into two main types, i.e., (i) exact algorithm and (ii) approximated algorithm. There are many existing studies based on approximation algorithm because VRP usually has high complexity and takes long time to solve. The approximation algorithm is usually known as a heuristic algorithm. However, this thesis focus on different aspects, which are modelling and application aspects, not algorithm, and thus the brief review of algorithm is presented in Section 2.6. Note that the experiments showed in this thesis were solved by a commercial solver named CPLEX.

At the end, the comparison of our studies with previous works is discussed in Section 2.7.

2.2 Types of vehicle routing problem

Most of existing VRP studies concentrate on ground-based vehicle delivery, i.e., by motorcycle, car, and truck, as they are convenient to be accessed. This section presents the problem statement of a traditional VRP with capacity and the extended VRP with diverse classes.

2.2.1 Traditional routing problem and its formulations

According to the paper [20], the authors defined the traditional VRP with capacity as the formulations in (2.1) to (2.9). The problem objective is to minimize the delivery cost for a shipper. The packages are distributed from a single depot to n number of customer addresses by multiple vehicles. Let $\mathcal{T} = \{T_1, T_2, \dots, T_{t'}\}$ denote a set of delivery vehicles, where t' is the total number of vehicles. Let $\mathcal{C} = \{C_1, C_2, \dots, C_{c'}\}$ denote a set of customers, where c' is the total number of customers. Each customer has only one package. When customers have two or more packages, the packages can be grouped together as one package. The package weight of customer i is defined as a_i . The delivery vehicles may have different capacities. Let b_t denote the capacity of vehicle t . All vehicles are required to start and end their journey at the depot. Let $\mathcal{U}$ denote a set of locations involved in the problem, where $\mathcal{U} = \mathcal{C} \cup \{depot\}$. The cost of traveling from location i to location j is denoted as c_{ij} , where $i, j \in \mathcal{U}$.

VRP with capacity constraint was formulated as a mix-integer programming (MIP). Two binary variables and one integer variables are used in this problem.

- X_{it} is a vehicle allocation in which $X_{it} = 1$ if vehicle t is allocated to customer i , and $X_{it} = 0$ otherwise, where $i, j \in \mathcal{C}$.
- $V_{i,j,t}$ is a routing variable in which $V_{i,j,t} = 1$ if vehicle t will travel from location i to location j , and $V_{i,j,t} = 0$ otherwise, where $i, j \in \mathcal{U}$.
- S_{it} is an auxiliary variable for eliminating a subtour in the routing solution.

Note that, in some publications, VRP is presented and explained as a mathematics graph instead of optimisation formulation.

Minimize:

$$\sum_{i \in \mathcal{U}} \sum_{j \in \mathcal{U}} \sum_{t \in \mathcal{T}} c_{i,j} V_{ijt} \quad (2.1)$$

Subject to:

$$\sum_{i \in \mathcal{C}} a_i X_{i,t} \leq b_t, \quad \forall t \in \mathcal{T} \quad (2.2)$$

$$\sum_{t \in \mathcal{T}} X_{it} = 1, \quad \forall i \in \mathcal{C} \quad (2.3)$$

$$\sum_{i \in \mathcal{U}} V_{ijt} = X_{it}, \quad \forall j \in \mathcal{C}, \forall t \in \mathcal{T} \quad (2.4)$$

$$\sum_{j \in \mathcal{U}} V_{ijt} = X_{jt}, \quad \forall i \in \mathcal{C}, \forall t \in \mathcal{T} \quad (2.5)$$

$$S_{it} - S_{jt} + nV_{ijt} \leq n - 1, \quad \forall i \in \mathcal{C}, \forall j \in \mathcal{C}, \forall t \in \mathcal{T} \quad (2.6)$$

$$X_{it} \in \{0, 1\}, \quad \forall i \in \mathcal{C}, \forall t \in \mathcal{T} \quad (2.7)$$

$$V_{ijt} \in \{0, 1\} \quad \forall i \in \mathcal{U}, \forall j \in \mathcal{U}, \forall t \in \mathcal{T} \quad (2.8)$$

$$S_{it} \in \{1, 2, \dots, c'\}, \quad \forall i \in \mathcal{C}, \forall t \in \mathcal{T} \quad (2.9)$$

The constraint in (2.2) ensures that the capacity is not exceeded. The constraint in (2.3) ensures that every customer is served by one vehicle. The constraints in (2.4) and (2.5) control the routing solution. The constraint in (2.6) is a subtour constraint. It eliminates disjoint tour in the solution. Lastly, the constraints in (2.7) to (2.9) indicate possible values for the decision variables. Note that, the subtour constraint in the paper [20] was adapted for easy understanding. The problem also can be formulated without variable $S_{i,t}$ by replacing the constraint in (2.6) by the constraint in (2.10).

$$\sum_{i,j \in \mathcal{S} \times \mathcal{S}} X_{i,j,t} \leq |\mathcal{S}| - 1, \quad \mathcal{S} \subseteq \mathcal{C}, 2 \leq |\mathcal{S}| \leq |c'|, \forall t \in \mathcal{T} \quad (2.10)$$

2.2.2 Variety of the existing vehicle routing problem

Since the VRP can be applied to vast scenarios in planning, scheduling, and other problems, many classes of VRP have been proposed. In this chapter, VRP is classified based on the problem characteristics and their application scenarios similarly to reasonable classifications in [136] and [20]. However, only relevant, critical, and

common VRP classes are described due to wide research area in VRP. These VRP classes are the combination of the traditional VRP with one or more additional characteristics and constraints. In this thesis, these classes are referred to as an extended VRP. At the end of this section, readers will have wide idea of multiple VRP classes and their characteristics.

VRP with Time Windows, a customer may have a specific delivery time that they want a package to be delivered, which can be called as a time window. There are two types of time windows in the literature, i.e., hard-time window and soft-time window. For hard-time window, the shipper must deliver a customer's package only within the requested time. For soft-time window, the shipper can deliver the package outside the request time, but the shipper needs to pay for the incurred penalty. Unlike the traditional VRP, the survey [20] presents the formulations of VRP with soft-time windows, and it also shows that many VRP with soft-time windows researches consider the re-loading process because when applying time-windows constraints, the re-loading condition appears as a bonus. The re-loading process is when a vehicle travel back to depot and load a new set of packages after the vehicle delivered all packages in the previous round.

In 1988, the authors in [94] presented the survey of the early VRP with time windows (VRPTW). The definition of hard-time window and soft-time window were defined, as well as the mathematics formulation of VRPTW. The single and multiple traveling salesman problem, the shortest path problem, the minimum spanning tree problem, the generic VRP, and the pickup and delivery problem combined with time windows constraints are discovered. Since scheduling problem of VRPTW creates high complexity, many heuristics algorithms and other techniques were proposed. Those algorithms and techniques were reviewed by the authors in [103] and [104] as continuing surveys with part one and part two in 2005. In the first part, the authors in [103] examined route construction and heuristics algorithms, i.e., traditional heuristics and local search. The second part presented the meta-heuristics, which had been used in VRPTW.

Pickup and Delivery Problem was first carried out in [52], as dial-a-ride problem. Many researches problem (i.e., backhauls, VRP with pickup and delivery, dial-a-ride) have been conducted relatively close to pickup and delivery

FIGURE 2.2: Classification of pickup and delivery problem [20].

problem, thus those problems are referred to be sub-classes of pickup and delivery problem. The survey about pickup and delivery problem is presented in [34]. According to the reasonable classification in [34], the authors in [20] summarised it into a figure as shown in Figure 2.2.

Multi-depots VRP (MDVRP) considers more than one depot, where each customer is visited once by a vehicle that departs from and returns to one of those depots. This class of VRP was first introduced in [29]. The optimisation techniques and formulations are demonstrated in [65]. Researchers applied MDVRP with many other classes, for example, MDVRP with time windows, MDVRP with Backhauls, MDVRP with pickup and delivery etc. The mentioned combinations also appeared as a small subsection in [20].

Location Routing Problem (LRP) is to help the shipper decide (i) whether the opening of one or more new depots is worth or not and (ii) the route of vehicles to serve customers. The objective is to minimize the cost of delivery and the fix cost of opening new depots. The early proposed LRP publication is [18]. Later on, the authors in [50] and [43] proposed the summarised review of LRP.

Generalized VRP is to simplify the VRP by allowing the shipper delivers packages to some locations, where multiple customers may pick up their packages there. Generalized VRP is to help the shipper decide (i) how to group customers together, and thus all the parcels of the group members are delivered to only one location, and (ii) plan the route of delivery vehicles. Multiple generalised VRP research have been proposed as the recent survey presents [108] in 2010. The authors in [108] also presented the combination of Generalised

VRP with traveling salesman problem with profits, VRP with selective backhauls, windy routing problem, and other problems.

Multi-compartment VRP considers the scenario that the shipper has multiple storage rooms in each delivery vehicle. A customer may require the shipper to deliver multiple types of parcels. Note that a customer demand cannot be split or grouped together as they may be in different types. This scenario usually occurs in many industries, e.g., food distribution (i.e., frozen and normal foods) and fuel distribution (i.e., diesel, benzin, gasoline). The optimisation model of the multi-compartment VRP with two compartments in one truck was proposed in [26]. Later on, one of the heuristics algorithms was examined in [1]. The overview of the multi-compartment VRP is presented in [137].

Multi-echelon VRP allows shippers to transfer customer packages to an intermediate depot before delivering packages to end customers. This problem addresses a long distance scenario, e.g., the depot of the shipper and customer locations are in different provinces. The objective is to minimize the delivery cost of all vehicles in all intermediate levels. One of the most common model is a two-echelon VRP, where two levels are considered. The first level is the link between a main depot to the second level, which consists of multiple intermediate depots, and the second level links those depots directly to the end customers. Since two and more echelon VRP are complicated problems and time consuming, the authors in [45] proposed two math-based heuristics to enhance the computational time performance. Later on, the authors in [99] and [131] proposed (i) the joint multi-echelon and time window VRP and (ii) the joint multi-echelon and heterogeneous VRP, respectively.

Heterogeneous VRP considers (i) multiple delivery modes, (ii) diverse characteristic of vehicles, or (iii) the combination of them, in the routing problem. The examples of delivery modes are discussed in chapter 1. The examples of the diverse characteristic of vehicles are traveling speed, equipment, and cost structure. The literature on heterogeneous VRP is classified based on the delivery modes as presented below.

1. **Ground-based delivery mode:** Recently, the authors in [136] and [20] reviewed heterogeneous VRP in brief. The surveys in [136] and [20]

refer heterogeneous VRP as “heterogeneous fleet” and “fleet size and mix VRP”, respectively. However, both of the reviews in [136] and [20] cover diverse characteristic of vehicles in only one dedicated delivery mode, i.e., ground-based delivery mode.

According to the survey in [20], heterogeneous VRP has been applied to multiple extended VRPs such as heterogeneous VRP with time windows, heterogeneous VRP with multi-depots, heterogeneous VRP with the combination of multi-depot and time windows. While the authors in [20] concentrate on the application aspect of heterogeneous VRP, the authors in [136] reviewed the algorithms (i.e., exact and heuristics) and the performance.

- 2. Aerial-based delivery mode:** The literature on aerial-based delivery mode in VRP can be referred to drone delivery mode. As this thesis focuses on drone delivery, the literature review of drone routing and delivery is presented as an individual section, which can be found in Section 2.3.
- 3. Sea-based delivery mode:** The recent review of sea-based delivery mode was presented in [82] as a ship routing. The authors in [82] also discussed the difference between sea-based delivery and the other modes, especially ground-based delivery, i.e., trucks, are listed as follows.
 - Ports’ fees must be paid when ships are at the ports.
 - The compatibility between a ship and a port must be considered, e.g., the weight of the load and ship draft.
 - Sea-based delivery usually involves multiple national trades.
 - Ships can be diverted at sea
 - Sea-based delivery usually takes a long period, e.g., weeks, and the multiple time windows may be covered by one ship for one stop.

Some of the major objectives are minimize costs [69][71][110], maximize profit [47][116], maximize the probability of winning campaign [83], and minimize the number of tanks [33]. Some of the major decisions are based on fleet size [47][83][33][69][71], deployment [47][116], design of transport system [55][70], and resource management. Multiple modeling methods and algorithms were applied, i.e., linear programming [33], integer programming [47][83], dynamic programming [69][71], stochastic

programming [116], heuristic algorithms [110], and simulation method [55][70]. More detail of sea-based delivery can be found on the related surveys in [82] [53].

4. **Outsourcing delivery mode:** This thesis always considers the outsourcing delivery mode with another delivery mode because outsourcing delivery is for a shipper outsources some packages to a third-party carrier, where the delivery cost is likely to be more expensive than using the shipper facilities.

Vehicle Routing Problem with Private fleet and common Carrier (VRPPC), which can be referred as a joint delivery between ground-based mode and outsourcing mode, was firstly introduced in [95]. The VRPPC was formulated using mathematical programming to obtain the solutions regarding (i) which size of the truck will be applied, (ii) which customer should be served by the truck or an outsourcing carrier, and (iii) what the routing path of the truck will be. To obtain the solutions, the greedy algorithm is adopted. The VRPPC [95] has been extended since then, e.g., [19], [81], [58], [60].

The authors in [19] formulated VRPPC by using integer programming [19]. They added initial cost of using the truck delivery and cost of using the outsourcing delivery mode into objective function. The decision variable of the outsource carriers was added into the optimisation model, and the outsourcing decision variable was solved together with truck routing variables. They proposed a heuristic algorithm, called TL-LTL algorithm to obtain the solution.

The authors in [81] proposed the improved heuristic algorithm of [19], called “Selection, Routing and Improvement (SRI)”. The authors in [81] used λ -interchange procedure in the result improvement step while the authors in [19] used three procedures, i.e., intra-route two-exchanges, inter-route one-exchange and two-exchanges. The results from SRI algorithm are always closer to optimal values than TL-LTL results.

Later on, the authors in [58] introduced the Iterated Density Estimation Evolutionary Algorithm (IDEA). It was shown that IDEA performs better than SRI. The comparison between genetic, SRI, and TL-LTL algorithms were presented in [60]. The evaluation indicated that under

various settings the genetic and SRI algorithms can reach optimal results more frequently than the TL-LTL algorithm. Furthermore, the genetic algorithm obtains a solution faster than the TL-LTL algorithm.

The detail of diverse VRP classes, which is described earlier in this section, has confirmed that VRP has a long history and a wide area in terms of problem types. The review of heterogeneous VRP has shown that it is one of the emerging topics and have several open issues that need to be addressed. Firstly, the aerial-based delivery mode in the heterogeneous VRP is an emerging topic and is new to package delivery industries. There are very few studies on the aerial-based delivery mode. As aerial-based delivery is one of the problems studied, more review is presented in an individual section (Section 2.3). Secondly, the existing studies on the outsourcing delivery mode in the heterogeneous VRP mainly focus on solution techniques, i.e., proposing or improving heuristic algorithms. Existing studies on outsourcing delivery mode ignore several modelling aspects, e.g. uncertainty formulations and delivery time windows formulations. By investigating on the heterogeneous VRP, [Objective 1](#) is addressed.

2.3 Detail of aerial-based routing and delivery

Recently, drones have been introduced as a great potential to be used in businesses. A number of researchers and engineers studied and improved the technical aspects of drones, i.e., enhance endurance, safety, and features. For drone-related research, most of the existing works consider surveillance applications. Only few works are dedicated to package/parcel delivery [72]. Therefore, this section presents the literature reviews of drone routing (in Section 2.3.1) together with drone delivery (in Section 2.3.2).

2.3.1 Drone routing

Recently, drone planning and scheduling problems have been studied in [16], [73]. The authors in [16] foresaw that drones could be used as mobile wireless communication nodes. In particular, the drones fly to collect data and return to transmit the data to the base station. The authors formulated the problem based on the

pick-up and delivery VRP and added the communication constraints, which were not considered in the previous works. The authors in [73] addressed the drone path planning problem, which is also one type of VRP, by minimizing the total drone resource requirement, i.e., fuel or battery. In the problem, all the customers need to be visited once. The drone is assumed to depart from a depot and visit multiple locations before returning to the origin. The drone is allowed to stop over at different locations to refuel or recharge its energy. The authors adopted resource constraint and developed a heuristic algorithm for the problem. Even the above studies are considered as VRP problem, the proposed scheme is dedicated for surveillance and communication applications, and thus most of their formulations are not suit for parcel delivery.

2.3.2 Drone delivery

Drone has emerged as an alternative package delivery mode. This is due to the fact that the reliability has been improved while the cost can be reduced substantially. As such, the VRP is being considered again for the drone delivery.

The authors in [11] considered the heterogeneous VRP for package delivery using both trucks and drones. The authors proposed the Flying Sidekick Traveling Salesman Problem (FSTSP) and the Parallel Drone Scheduling Traveling Salesman Problem (PDSTSP). In the FSTSP, the truck driver can set up a drone and let it visit different customers. After visiting and dropping packages to the assigned customers, the drone returns to the truck. Hence, the truck is referred to as a mobile depot for the drone separation and re-connection. On the other hand, in the PDSTSP, the authors formulated the problem with only one depot, but with multiple drones. Both the problems were formulated as a mix-integer programming model, where the objectives of the FSTSP and the PDSTSP are to minimise the serving time for both vehicles, i.e., one truck and one drone, and to minimise the traveling time of the truck, respectively.

The authors in [119] also considered the joint truck and drone delivery problem. They proposed mathematical formulations for closed-form estimations. The K-mean algorithm and genetic algorithm are used to obtain the delivery solution.

However, the previous studies in [11] and [119] concentrate on developing a solving algorithm. The authors in [148] presented the worst case scenarios from the theoretical point of view when using hybrid drone and truck delivery. They considered the case that drones can depart and arrive at truck at a depot or a customer location, which means that the drones can pickup a customer's package from the truck. They derive a number of worst-case results, which depend on the number of drones per truck and the speed of the drones relative to the speed of the truck.

Later on, the authors in [72] considered only drones for package delivery planning. They proposed two drone routing optimisation problems given the constraints on capacity, battery weight, charging and payload weight. The authors modelled multiple drones, and each drone is able to serve multiple customers before returning to the depot. Drones are also allowed to come back to refuel or charge a battery at the depot for multiple times. The objectives are (i) to minimise the cost subject to delivery time constraint and (ii) to minimise the time subject to the budget cost constraint. Then, they proposed the simulated annealing (SA) heuristic algorithm to address the multi-trip drone routing problem.

The authors [54] also studied the drone-only package delivery problem. Unlike other previous studies that assume drones to fly only over roads and street, the problem focuses on the flying direction of drones, which can avoid barriers and obstacles automatically. The problem is formulated as mix-integer programming, and a heuristic algorithm is used to find the solution.

To sum up, all the above studies, i.e., [11] [119] [148] [72] [54], are dedicated for a package/parcel delivery application.

2.4 The modelling characteristics of optimisation parameters and objective function

VRP for package delivery can be categorised as a sub-problem of package delivery planning. Again, VRP is dedicated to determine vehicle route for a single shipper. In package delivery planning, multiple shippers can be considered. The approaches for multiple shippers are presented later in Section 2.5.

This thesis divides the modelling delivery planning for a single shipper based on optimisation components (i) parameters, (ii) decision variables, (iii) objective functions, and (iv) constraints, which is shown in Figure 2.1. However, VRP is always a discrete and constrained problem. Therefore, this section presents only two optimisation components, i.e., (i) the characteristic of input parameters in Section 2.4.1 and (ii) characteristic of objective functions in Section 2.4.2. Note that all the references in this section are dedicated to a single shipper.

2.4.1 Uncertainty in optimisation parameters

The VRP has been analysed from different aspects, and the VRP with uncertainty is one of the major aspects [87] [143]. For more than 30 years, VRP with uncertainties have been investigated as it is unavoidable in many situations.

TABLE 2.1: Classification of problems with uncertainty

	Deterministic input	Random inputs
Inputs are known beforehand	static and deterministic	static and stochastic
Inputs are changed overtime	static and dynamic	dynamic and stochastic

The VRP with uncertainty can be divided into two types, i.e., non-anticipative and anticipative. The former is only react to updates based on the new data, and the latter considers knowledge on the new data (e.g. probability) to predict the future [141]. In general, the non-anticipative method is designed for a dynamic and deterministic problem, while the anticipative method is developed for a static and stochastic (also known as offline stochastic) problem. The difference between dynamic and stochastic are explained in Table 2.1. Note that the traditional VRP, which is indicated in Section 2.2.1, does not take any random parameters into account, all parameters are assumed to be known before solving the problem, and thus the problem is classified to be a static and deterministic problem based on Table 2.1. Moreover, there are existing studies of mix anticipative and non-anticipative method, which can be referred to be a dynamic and stochastic problem, but they are very rare according to the survey in [140].

This subsection presents the literature reviews of (i) stochastic VRP, (ii) dynamic VRP, and (iii) the combination of stochastic and dynamic VRP as follows.

2.4.1.1 Stochastic VRP

A problem is called stochastic VRP when one or more parameters are random and the delivery solution is achieved before performing the delivery, i.e., a scenario tree solution is achieved to handle the uncertainty of the random parameters.

Since a better solution can be achieved when uncertainty are taken into account, stochastic VRP have gained high attention and have been a fast growing research area. The random parameters which are usually considered in stochastic VRP are ordering demands, the quality of demands, customer addresses, traveling time. Multiple stochastic VRP have been proposed in diverse categories, i.e., VRP with stochastic demand [84], VRP with stochastic demand and traveling time [96], VRP with stochastic traveling time and serving time [147], and multi-compartment VRP with stochastic demands [56].

The authors in [89] proposed the literature review of stochastic VRP. The comparison between chance constrained program (CCP) and as a stochastic program with recourse (SPR) were presented in [89]. However, they admitted that only a relatively small size of stochastic VRP can be tractable. Exact algorithms and good heuristics algorithms are hard to design when randomness is involved. Later on, the authors in [88] decomposed the VRP with stochastic demands and customers into multiple sub-problems by L-shape method [37]. By decomposing, the problem became more tractable.

According to the survey [89], many stochastic VRP has been proposed. However, there are very few studies related to drone delivery. To the best of our knowledge, none of stochastic drone delivery has been proposed. The review of stochastic drone orienteering problems are presented in the next paragraph. In particular, a solution of the orienteering problem is to visit only some locations while the VRP requires all locations to be visited.

A stochastic drone mission planning is presented in [76] and [75]. The authors in [76] proposed a drone planning model with the stochastic resource consumption, i.e., fuel or energy consumption, traveling from one location to another location. The authors presented the comparison between the robust solutions and the solution with average resource consumption. Instead of considering stochastic resource consumption, the authors in [75] considered the uncertainty in recording time and

traveling time from one location to another location, as well as a visiting time interval of each location, which is known as time-windows. The authors combined two objectives together by adding and balancing two weight parameters. The objectives are to maximise (i) the expected profits of visiting the locations which are already known before taking off and (ii) the percentage of reaching new locations, which are not known before taking off, on time. The new locations are emergency and unplanned cases. The authors proposed a re-planning approach as integer programming and developed a heuristic algorithm to solve it.

2.4.1.2 Dynamic VRP

A problem is called as Dynamic VRP when the values of one or more parameters are changed over the execution time of delivery, which can be referred as an evolution of information. Dynamic VRP also can be called as online or real-time VRP. After reviewing many papers on VRP, I have found that some studies proposed dynamic VRP, but call the parameter as stochastic parameters, which likely to mislead readers to thought that the problems are stochastic VRP.

Dynamic VRP has been investigated for many years. One of the early works were published in 1980 [51] to model and solve pickup and delivery VRP with random demand. The system were formulated and solved as a dynamic programming, which customer demands are revealed over time. Later on, many researches have combined dynamic VRP with extended VRP (i.e., VRP with time windows, pickup and delivery VRP) as presented in [142] [20]. Recently, the authors in [142] proposed the comprehensive review of dynamic VRP, which concentrated on both algorithms and applications. The authors in [142] divided the dynamic and deterministic VRP as two categories as follows.

Periodic reoptimisation was first proposed in [51] as a dynamic programming approach. The main focus is to found an optimal solution each time obtaining a new request. Later on, the author in [144] presented the drawback of this approach, which was less scalable of application instances. The author named it as “curse of dimensionality”.

The authors in [67] proposed the dynamic or real-time point to point delivery problem with a strong assumption that one truck can serve one request at

a time. They proposed a rolling horizon approach based on a linear program, and the linear programming was re-solved every time a new request arrives. The authors in [156] a column generation approach for dynamic VRP. Their contribution is to use columns from the previous decision to generate columns for a set-partitioning model. Later on, the author in [93] presented the comparison between the approach in [156] and the traditional column generation with no time limit. The evaluation shown that the approach in [156] achieved comparable results in terms of objective function, but the computational times of the approach in [156] were much faster, i.e., approximately more than 360 times faster.

Moreover, the authors in [113] implemented ant colony optimisation to address dynamic VRP. They divided one day time into multiple periods. Similar approach can be found in [77] and [3].

Continuous optimisation mainly bases on an adaptive memory. the original route plan is changed based whenever the available data changes. The performing trucks do not know the next customer until they finished to serve the current customer. Dynamic VRP with continuous optimisation was early introduced in [86]. The authors in [86] extended the study in [27] by implementing a parallel tabu search. By using multiple-thread programming technique, they obviously obtained faster computational time which is suitable to be used in real industries. When the shipper receives a customer request, the system helps to decide whether to accept or reject the request. In additional, the authors in [86] introduced a generalisation of the tabu search with adaptive memory, which was named as Multiple Plan Approach. From the aforementioned studies, I can found that most of the researchers, who worked on the dynamic VRP with continuous optimisation, focused on heuristic solver.

In 2019, the author in [150] proposed a integer programming to address the dynamic drone delivery for meal. The proposed framework in [150] can be categorised as a periodic reoptimisation. The author proposed a integer programming with an algorithmic progression. CPLEX was used to solve the integer programming while the proposed algorithms looped over and called the proposed optimisation as an inner function. They considered practical factors, i.e., the orders' locational uncertainty, food type restriction, order size variation, drones' mobility range, carrying

capacity, battery consumption and battery swapping operation in the optimisation model. At the end, the experiments were performed on the simulations of case studies.

2.4.1.3 Stochastic and dynamic VRP

The survey [140] shows that few existing works consider both dynamic and stochastic problems simultaneously. Almost all of them are dedicated for the VRP with one random parameter, e.g. customer location, customer demand, or traveling time.

The authors in [78] proposed a model to solve the dynamic and stochastic VRP with a sample scenario hedging heuristic. They considered the case that customers' addresses and demands are known at the same time. They formulated the model as a multi-stage stochastic optimisation. Each stage is separated from the other stages by the event that new requests are generated. In their system formulation, the first stage and the second stage decision variables are not decided at the same time. This means that the solution of the first stage is not based on the second stage. Later on, the authors in [141] considered the dynamic and stochastic VRP with random customers' locations and demands as in [78]. The authors in [141] proposed a new framework, which is java implementation of the multiple scenario approach. However, they only focused on the heuristic solver. In addition, the authors in [46] and [133] also focused on modelling and formulating the dynamic and stochastic optimisation problem for logistics application. They obtained their experiment results from CPLEX solver. However, their main goal is on the container allocation, not vehicle routing. Recently, the re-route delivery planning system were proposed with two components, i.e., the stochastic optimisation and the re-route algorithm in [121]. New dependency constraints and capacity constraints were also proposed. Unlike the above works, the study in [121] focus on defining reasonable scenarios and realistic system model as well as developing problem formulations, i.e., an exact method.

This subsection has presented the review of modelling based on the characteristics of input parameters. Next, I present the literature review of modelling which is classified as a multi-objective optimisation.

2.4.2 Multiple objectives for VRP

Multi-objective models for the vehicle routing problem (VRP) has been introduced, and the review of multi-objective VRP can be found in [102]. Although there have been a lot of studied on the drone delivery problem [72] [123] [122], they are commonly modelled as a single-objective optimisation, despite the fact that industries may consider the drone delivery problem as a multi-objective problem in nature.

However, to the best of our knowledge, only the authors in [107], [30], and [8] considered the multi-objective problem for the drone delivery according to the survey in [102]. The difference of the studies in [107], [30], [8], [49], [149], and the proposed MODD are summarised in Table 2.2.

The authors in [107] addressed the multi-objective drone delivery problem by transforming the decision variables to a single cost value by using the Multi-Criteria Decision Making (MCDM) method. Then, the grid-based search algorithm, i.e., A*, was used to solve the problem. The authors in [30] proposed a multi-objective modelling for drone VRP with soft time-windows constraints. The optimisation aims to (i) minimise the traveling distance, (ii) minimise the number of used drones, and (iii) maximise the customer satisfaction, which decreases every time step when the drone cannot reach the assigned location on time. The exact method, i.e., ε -constraints, was used to address the multi-objective optimisation. However, The authors experimented the proposed system with a filming football match application instead of package delivery applications. The authors in [8] proposed an optimisation with several objectives. They transformed the multi-objective optimisation into a single-objective optimisation by adding weight parameters to all the decision variables of the objectives, and used the multi-objective smart pool search (MOSPOOLS), which is a matheuristic algorithm, to determine the weight parameters.

Note that the the studies in [49] and [149] were published in 2020.

TABLE 2.2: The comparison of the related work and my study

	Multi-objective	Solver	Time involve	Stochastic event	Application
[107]	multi-criteria decision making	gird based search algorithm (A*)	min time to destination	-	UAV Parcel delivery.
[30]	ϵ -constraint	MILP CPLEX	time window constraints	-	VRP for UAV, but the model is dedicated for VRP with set of Point of Interests (PoIs), i.e., data collection.
[8]	multi-objective smart pool search	MILP CPLEX	min time to destination	-	UAV Parcel delivery. The location and route are considered as x and y coordinates.
MODD	ϵ -constraint	MILP CPLEX	time window constraints	take-off and breakdown conditions are considered	UAV Parcel delivery. The MODD framework considers locations and route similar to traditional VRP, which distance matrix is given.
[49]	ϵ -constraint	MILP CPLEX	time window constraints	-	UAV Parcel delivery. The study focuses on sustainability concept e.g. minimizing carbon emission and maximizing total sustainability score.
[149]	Push-Forward Insertion heuristic, local search	artificial colony algorithm	time window constraints	-	Joint truck and drone parcel delivery. The Objectives are to (i) minimize the truck power consumption, (ii) minimize the drone power consumption, and (iii) minimize the number of trucks.

2.5 Modelling delivery planing for multiple shippers

Since the cost is a sensitive factor in logistics industry, partnership and cooperation among shippers are seen as a viable solution to improve resource utilisation or trade transportation requests among cooperative shippers. Section 2.4 addresses problems in different aspects, and those approaches alone are not suitable to handle multiple shippers. In this section, a review of the cooperation approach in VRP and related applications are presented. The related works are categorised based on approaches, i.e., game theory and auction. The review of the approaches are explained as follows.

2.5.1 Game theory approach

The cooperative game have been used to help shippers decide how to form a co-operation group. The authors in [80] introduced the joint routing and cooperation among shippers for ground-based package delivery. They used the Shapley value to fairly distribute cost among shippers. Later on, the cooperative game theory was used in many ground-based package delivery studies [106], [85], and [2]. However, all of them concentrate on presenting different methods for cost sharing. The review of cost sharing for shipper cooperation in package delivery was presented in [90]. The authors in [90] also reviewed multiple types of transportation system, i.e., traveling sale man problem (TSP), vehicle routing problem (VRP), Joint distribution problem, and inventory problem. They found that Ad-hoc method and Shapley Value method are the most popular cost sharing method which had used in the related works. They defined an Ad-hoc method as a method which proposed for a particular context. A commonly found approach in Ad-hoc methods is to find core allocations. Note that Shapley Value was referred as a traditional cost sharing method in [90].

Recently, the authors in [112] proposed a framework for shipper cooperation for the capacitated vehicle routing problem with time window. Instead of focusing on the cost sharing methods, they considered the overlapping coalition formation among shippers, i.e., one shipper may join multiple resource pools at the same time. The

Myerson value was adopted as the solution to distribute cost among cooperative shippers.

To the best of our knowledge, the study of the cooperative among shippers in drone delivery has not been proposed yet. Especially, the uncertainty of behaviour of the cooperative shippers can directly effect to the delivery cost and the satisfaction of customers, but this uncertainty in the shipper behaviour has not been studied yet.

2.5.2 Auction approach

Trading transportation requests in parcel delivery can be done by using auction based exchange system. Integrating auction mechanism with the optimisation, e.g. VRP, a delivery planning for shipper cooperation can be achieved. Most common exchange mechanism allows shippers to bids on groups of customer packages, i.e., bundles, rather than an individual package [21][36]. On the other hand, the authors in [92] allows shippers to bids on an individual package, and handled the interdependence between packages by using look-ahead strategies for bid pricing and bid evaluation. In addition, they considered the cases that customer requests, i.e., package, arrive one by one, which can be referred as a dynamic problem.

The authors in [21] proposed an optimisation model, which helps a shipper decides (i) whether a bundle of customer packages is taken or not, (ii) the price of the bundle, and (iii) the routes of involved trucks. the uncertainty of bidding process was taken into account. Furthermore, two heuristic algorithms was proposed to solve the proposed model. Later on, the authors in [36] also proposed a stochastic bid price optimisation model, which the major objective is to maximise expected profit. However, the authors in [36] did not consider the routing problem into the system while the authors in [21] considered. They included only constant costs of serving each bundle. Therefore, shippers need to calculate the cost of serving before performing their proposed system. Similar to the study in [21], the evaluation was performed by using a heuristic algorithm, i.e., the proposed iterative coordinate search algorithm.

The authors in [115] proposed an exchange mechanism by implementing a Vickrey auction and a combinatorial auction. Later on, the authors in [91] proposed a framework which help the shippers (i) decides which customers to be served by the

original shipper and which customers will be offered to other shippers (ii) generate bids reflecting their willingness to buy customer requests from other shippers and (iii) decide the vehicle route for each shipper. For auction exchange system, they adopted a framework from [115], which has two types, i.e., (i) Single request reassignment (SRRA) and (ii) Bundle request reassignment (BRRA). For vehicle routes, they proposed a pickup and delivery problem as an integer programming problem, but solving it by a heuristic algorithm when they performed the evaluation. Shippers can be both buyer and seller in the auction process. Multiple strategies for selecting customer requests was presented. However, the authors in [91] only presented the comparison between cooperative delivery, but did not present the improvement of cooperative method from the based line, i.e., individual delivery by each shipper.

To the best of our knowledge, the study of the cooperative drone delivery for multiple suppliers has not been addressed by any auction mechanism yet.

2.6 Solution techniques

Since VRP was introduced, hundreds of algorithms have been proposed to solve traditional VRP and the extension. VRP is considered as an NP-hard problem. Thus, the computational time becomes intractable when the size of the problem grows. In general, VRP algorithms can be classified into two types, i.e., exact algorithms and approximate algorithms. An exact algorithm usually guarantee to return the optimal result while it usually takes long time to solve and can handle uncomplicated problems. On the other hand, an approximate algorithm usually returns a nearly optimal result, i.e., local optimal, while the solving time is faster.

Exact algorithm: according to the surveys in [40] and [39], an exact algorithm can be classified to three main classes as presented in Figure 2.1.

1. Direct tree search method: the authors in [101] are one of the earliest researchers who discussed about VRP with branch and bound method. Later on, the authors in [114] extended the idea and proposed the branch and bound method for travel salesman problem and VRP. The 14 steps algorithms with the flow chart and a solution example were presented

in detail. The tree is built based on the reduced table, which represents the subtraction of distance table between a location to other locations. Vehicles were added to the distance table as virtual depots. Moreover, the authors in [40] reviewed the branch and bound method for VRP by explained (i) branching on arcs and (ii) branching on routes, which were extended from [101] [114].

2. Dynamic programming: Dynamic programming was early introduced by the authors in [117]. Dynamic programming uses recursion formulation to find the minimise delivery cost. The summarised of general dynamic programming for VRP and the formulations can be found in [40] and [39]. Later on the authors in [100] proposed state-space relaxation for dynamic programming. The authors in [40] reviewed that the state-space relaxation technique can reach the optimal cost faster than the general branch and bound method. Note that the branch and bound method usually relies on the computational time of every node on the search tree.
3. Integer linear programming (ILP): this category is considered to be a very board topic and the most attractive topic to researchers. The authors in [40] categorised it into three different sub-classes as follows.
 - Set partitioning formulations: there are two main decision variables in the proposed optimisation, i.e., (i) an indication of the route selection in which route r is the solution or not, and (ii) an indication whether path from location i to location j is selected for route r or not. To sum up, the set partitioning formulations build multiple routes and choose the solution based on the routes.
 - Vehicle flow formulations: a main decision variable was proposed for vehicle flow formulation in which vehicle t will travel from location i to location j or not. The traditional VRP, which proposed in Section 2.2.1, and our studies belong to this subclass.
 - Commodity flow formulation: it can be considered as an extension of the vehicle flow formulations because it contains the same decision variable plus one additional decision variable. The additional variable represents the total quantity of the carrying packages from location i to location j .

FIGURE 2.3: Solution quality and computational time performance of meta-heuristic methods [136].

The intensive reviews with formulations of all the sub-classes can be found in [40] [39]. The researchers who work on this ILP class focused on how to improve the formulations in VRP. For example, for vehicle flow formulation, the authors in [40] reviewed three- and two- index formulations, which the derived the formulations by aggregating three-index decision variables into two-index decision variables, i.e., $V_{i,j,t} = \sum_{t \in \mathcal{T}} V_{i,j}$. Note that the meaning of the mathematics notation can be found in Section 2.2.1.

Moreover, the authors in [39] discussed the definitions, formulations, and processes of the representative algorithms from these three classes extensively. However, the authors in [39] did not compare the difference between the three classes while the authors in [40] concluded that direct tree search offers the least potential for grown while they believed that dynamic programming can be expanded on state-space relaxation approach. The ILP seemed to be the most popular and promised to be most effective and successful class. However, The ILP is usually formed a tighter problem, and thus it can not be handle in the same way as a loose problem, which are the other two classes.

Approximation (meta-heuristic) algorithm: since a practical problem in industries is more likely to be a large scale problem and needs satisfactory computational time, many researches have been concentrated on proposing an approximate algorithm according to the surveys in [38][136]. Apparently,

TABLE 2.3: References and Acronyms of metaheuristics, which are indicated Figure 2.3

Acronym	Approach
VCGLR11s [135]	Hybrid Genetic Algorithm
VCGLR11f [135]	Hybrid Genetic Algorithm
NB09 [155]	Hybrid Genetic Algorithm
GGW11 [12]	Hybrid: Parallel and Neighborhood Centered Search
MB07s [24]	Hybrid: Evolutionary Algorithm + Local Search (slow)
ZK10 [28]	Hybrid: Local improvement heuristic + Tabu Search
JCL11 [61]	Tabu Search
MM11 [154]	Bees mating
JCL12 [62]	Hybrid: Parallel Tabu Search
P09 [15]	Hybrid: Greedy Randomise Adaptive Search Procedure + Local Search
RDH04 [98]	Ant colony optimisation
T05 [17]	Tabu Search
CM11 [59]	Hybrid: Parallel Tabu Search
MM10 [153]	Hybrid: Genetic Algorithm+ Particle swarm optimisation
DK07 [138]	Guided local search
MB07f [24]	Hybrid: Evolutionary Algorithm + Local Search (fast)
PR07 [25]	Adaptive Large Neighborhood Search
LGW05 [31]	Local improvement heuristic
MMP06 [152]	Hybrid Genetic Algorithm
P04 [14]	Hybrid Genetic Algorithm

in many studies, the more time we save to solve an optimisation problem, the larger gap between the optimal and the achieved solutions will be.

Recently, the authors in [20] summarised the approximate algorithm as presented in Figure 2.1 (heuristic algorithm node). They concluded that metaheuristics consistently achieves a nearer optimal solution and a faster computational time when comparing to classical heuristics. Moreover, the authors in [136] presents the overviews of heuristics and meta-heuristics, and 64 remarkable meta-heuristics were intensively reviewed. The authors in [136] summarised the performance of heuristic algorithms in term of both computational time and the gap between the best known solution (optimal) and the achieved solution as indicated in Table 2.3 and Figure 2.3. From Figure 2.3, the proposed algorithms in [24] and [135] achieved the fastest solvign time and the smallest gap, respectively. However, the algorithm in [135] performed nearly 100 times slower the algorithm in [24]. In term of accuracy, i.e., gap, the algorithm in [135] achieved only 0.2%, which is about 6 times better than

the algorithm in [24]. Therefore, the best algorithm for each shipper can be difference depended on his/her preference.

2.7 The summarise of the research gaps

This section presents the summary of the research gaps. Firstly, the stochastic optimisation for a single shipper is presented. The cooperative resource planning for multiple shippers is discussed later.

2.7.1 Stochastic optimisation for a single shipper

The existing studies of drone delivery, which are discussed in 2.3.2, do not consider stochastic parameters in the drone delivery problems. Stochastic VRP has been introduced and played a significant role in logistics research for many years [89]. As unpredictable situations in drone delivery can happen more frequently than a ground-based delivery and may cause delay or unsuccessful delivery, thus they will hamper customer satisfaction and business reputation. The unpredictable situation could be (i) drones cannot take off due to weather conditions such as raining and strong wind, (ii) drones may have technical problems during flying, and (iii) drones can have an accident such as a bird strike. Moreover, using drones to deliver packages may not be able to fulfil all customer requirements, e.g., customer address is outside of drone flying distance. The summary of the research gap in which a single shipper is considered is presented in Table 2.4.

In Chapter 3, I propose the joint ground and aerial delivery service optimisation and planning (GADOP) framework to improve the efficient of the merging between different delivery business models (Objective 1). The GADOP framework is formulated as a stochastic integer-programming problem to address the unpredictable situations in drone takeoff and breakdown conditions (Objective 2). With stochastic parameters, the delivery can be investigated and planned more effectively while trading with computation time.

Apparently, the size of the problem grows exponentially when a stochastic parameter is considered. Thus, it is difficult for an industry to solve the stochastic problem with an acceptable time. In the real world, shippers have limited time to

TABLE 2.4: The comparison between existing studies and the proposed frameworks when a single shipper is considered

Objective	Existing studies	Proposed frameworks in this thesis
Objective 1	<ul style="list-style-type: none"> - Joint ground-based and outsourcing package delivery planning was investigated in [19], [81], [58], [60]. However, the existing studies focus on improving the heuristic algorithms or solution techniques while the mathematical model and the formulations are still lacking. - Joint ground-based and arial package delivery planning is proposed in [11], but it considers the flying time as the objective function. In general, package delivery problem considers the delivery cost as the objective function and the flying time or delivery time is set as one of the constraints. - To the best of my knowledge, a joint arial and outsourcing package delivery planning has not been studied. 	<ul style="list-style-type: none"> - The GADOP framework is proposed to address the joint ground-based and arial package delivery planning. Extending from the study in [11], the model is reformulated by minimising the total delivery cost as the objective, and the delivery time for both trucks and drones are controlled by the constraints. - The MODD framework is proposed to address the joint arial and outsourcing package delivery planning.
Objective 2	<ul style="list-style-type: none"> - The existing drone routing and drone delivery do not consider any uncertainty parameters, e.g. takeoff and breakdown conditions. - As drone delivery is less reliable than ground-based delivery, uncertainty parameters are unavoidable. - To the best of my knowledge, currently none of the existing hybrid delivery model considers uncertainty parameter. 	<ul style="list-style-type: none"> - All the three proposed frameworks consider uncertainty parameters in arial package delivery planning. - Takeoff and breakdown conditions are considered to be uncertainty parameters in all the frameworks. - The GADOP and MODD frameworks are formulated as three-stage stochastic optimisations. Uncertainty parameters are observed between each stage. - The BCoSDD framework is extended from the GADOP framework by reformulating the problem to be a multistage stochastic programming optimisation.
Objective 3	<ul style="list-style-type: none"> - Most of the existing studies focus on reducing the computational time by using approximation algorithms, e.g. heuristic algorithms. - The study in [64] adopted L-shape method to reduce the computational time of a general stochastic VRP. The problem was decomposed into multiple sub-problems. As a result, the complexity of the problem was decreased. 	<ul style="list-style-type: none"> - The L-shape method is adopted in the GADOP framework to reduce the computational time. - When using L-shape method, the GADOP framework can be further extended to a heuristic algorithm to further reduce the computational time. - The GADOP framework is decomposed into three sub-problems. With the proposed formulations, the second-stage sub-problems and the three-stage sub-problems in the decomposed GADOP framework can be executed simultaneously. As a result, the decomposed GADOP framework can further adopt parallel computing technique to split the sub-problems to execute in multiple computing processors.

Objective	Existing studies	Proposed frameworks in this thesis
Objective 4	<ul style="list-style-type: none"> - Multi-objective VRP was considered in multiple studies [102]. - To the best of my knowledge, only [107], [30], and [8] consider multi-objective for aerial routing problem. - Their proposed multi-objective optimisation model do not consider uncertainty parameters. - The study in [107] uses multi-criteria decision making method to combine multiple objectives into single objective by using weighting parameters. - The study in [30] uses ϵ constraint method, but the problem type is a data collection problem and is not a package delivery problem. - The study in [8] uses smart pool search (heuristic) to handle multiple objectives. The problem type in [8] is to calculate the direct distance from x and y coordinates. 	<ul style="list-style-type: none"> - The MODD framework is proposed to address multi-objective package delivery problem with uncertainty parameters, i.e., takeoff and break down conditions. - The MODD framework is the extension of the study in [30] since the MODD framework uses ϵ constraint method to handle multiple objectives. - The MODD framework is dedicated to aerial package delivery planning, unlike the studies in [30] and [8].

plan their delivery, and the slow solving problems may be considered as unusable. Therefore, the simplification of delivery planning and routing is also a significant challenge. To reduce the computational time of solving the problem (Objective 3), I introduce a decomposition algorithm by adopting L-shape method [64] to the GADOP framework.

On the other hand, most of the existing studies formulated drone delivery problem as a single objective optimisation, despite the fact that shippers may consider drone delivery problem as a multi-objective problem (Objective 4). Industries may consider this as another challenge. Thus, I proposed the MODD framework to help a shipper, who considers multiple objectives, plan his/her delivery in 4. The MODD framework is the extension of the single objective optimisation in the GADOP framework.

The existing studies of multi-objective drone routing and delivery are presented in Section 2.4.2. The proposed multi-objective system in [30] addresses the case that drones can visit multiple locations before returning to the depot. However, in reality, a drone is likely to carry only one package at a time. Therefore, I reformulate the problem for the case that a drone carries one package to serve a customer and returns to the depot before serving the next customer. The authors in [8] converted multi-objectives into single-objective by adding weight parameters and using the multi-objective smart pool search to adjust the weight parameters. Instead of using the heuristic method as [8], I use the ϵ -constraints to perform the experiments with exact solutions. Again, the drones are less reliable than ground-based vehicles. Therefore, in 4, I propose a three-objective and three-stage

TABLE 2.5: The comparison between existing studies and the proposed frameworks when multiple shippers are considered

Objective	Existing studies	Proposed frameworks in this thesis
Objective 5	<ul style="list-style-type: none"> - The VRP with multiple shippers were studied in [80] [106], [85], and [2]. However, all of them concentrated on presenting different methods for cost sharing. - Shapley Value is one of the most popular cost sharing methods, and it is referred to be a traditional cost sharing method [90]. - The overlapping coalition formation for the ground-based package delivery planning with multiple shippers was recently proposed in [112]. - The existing drone delivery planning by optimisation is dedicated for a single shipper. - The literature also misses the study of joint optimisation package assignment, shipper co-operation, and cost management in drone delivery. 	<ul style="list-style-type: none"> - The BCoSDD framework is proposed to address the hybrid drone and outsourcing delivery planning problem when multiple shippers are considered.
Objective 6	<ul style="list-style-type: none"> - To the best of my knowledge, the uncertainty in shipper behaviour has not been considered to enhance the effectiveness of any package delivery planning. 	<ul style="list-style-type: none"> - In the BCoSDD framework, the uncertainty in shipper behaviour is considered and handled by Bayesian approach. - The belief update machine is proposed to learn the behaviour of the cooperative shippers.

stochastic optimisation to handle random parameters, i.e., takeoff and breakdown events, which is similar to the random parameters in Chapter 3.

2.7.2 Cooperative resource planning for multiple shippers

The comparison between existing studies and the proposed frameworks is summarized in Table 2.5.

Package delivery planning in VRP mainly focuses on the benefit of only one shipper, which can be known as supplier or agent. However, shippers may cooperate with other shippers to share their facilities and achieve a higher profit. In 2016 and 2017, the cooperation between multiple shippers in VRP was proposed in [129][112]. Both works adopted game theory method to the VRP with multiple depots [129] and stochastic heterogeneous VRP with outsourcing carriers [112]. It is essential to note that all previous studies of the shipper cooperation in package delivery are dedicated to the ground-based delivery as the literature review presented in Section 2.5. Therefore, this shipper cooperation is an area that can be further investigated. Moreover, the existing drone delivery planning by optimisation, i.e.,

linear or non-linear programming, are dedicated for a single shipper. To introduce a new methodology and a new algorithm to manage drone package delivery among shipper cooperation ([Objective 5](#)), the BCoSDD framework is proposed in Chapter 5.

In addition, in the real world, there is uncertainty in shippers' behaviour since the shipper cannot be certain about the other shippers with whom they cooperate. Some of the cooperators may or may not serve customers as they originally agreed. This uncertainty in shippers' behaviour is also one of the challenges. To the best of my knowledge, none of the existing studies considers shippers' misbehaviour. To learn and effectively plan based on the shipper behaviour ([Objective 6](#)), the BCoSDD framework is proposed with Bayesian approach. The literature also misses the study of jointly optimising package assignment, shipper cooperation, and cost management in drone delivery. Therefore, they are the focus of Chapter 5 in which the uncertainty in drone breakdown is also taken into consideration.

Chapter 3

Single Objective Optimisation for Truck and Drone Delivery ¹

This chapter introduces the joint ground and aerial delivery service optimisation and planning (GADOP) framework for a shipper. In particular, customers' packages can be delivered by trucks or drones. The GADOP framework considers the case that the shipper needs to make reservations for both trucks and drones in advance. The shipper, after deciding packages to be delivered by trucks, determines route for each truck. Similarly, the shipper determines package serving order of the drones. I model the uncertainty of the drone delivery in terms of takeoff and breakdown conditions. Specifically, the drone may not be able to take off, e.g., due to bad weather, and if it can take off, the drone may experience accident, e.g., due to technical issues. Therefore, the stochastic programming technique is adopted to formulate the joint ground and aerial delivery service optimisation and planning in which the decisions are made in different stages.

According to the objectives in Section 1.2, [Objective 1](#), [Objective 2](#), and [Objective 3](#) are accomplished in this chapter. Again, the contributions of this chapter are summarised as follows. Each contribution fulfils one of the objectives.

Objective 1: The proposed joint optimisation is able to utilise and trade off benefits and limitations of truck and drone delivery. In particular, the proposed

¹The work in this chapter has been presented in [\[123\]](#)

optimisation can achieve the minimum total payment² for the shipper by considering the routing path of trucks and the serving order of drones.

Objective 2: The formulated stochastic programming model can find and achieve an optimal solution taking the uncertainty of the drone delivery into account. Consequently, the shipper can obtain the better planning for the truck and drone usage given certain constraints such as capacity limit and traveling distance limit.

Objective 3: To reduce the computational time of obtaining the solution of the stochastic programming model, I introduce the decomposed GADOP by adopting the L-shape method. The decomposition allows the model to be split into multiple smaller sub-problems, which are more computationally tractable. The proposed decomposition method is summarised as a pseudo code in Algorithms 1.

This chapter is structured as follows. The system model and the formulations of the proposed joint optimisation are presented in Section 3.1 and Section 3.2, respectively. Section 3.3 presents the decomposition of the proposed joint optimisation. At the end, Section 3.4 presents the experiment setting and the numerical results of the proposed joint optimisation.

3.1 System model and assumptions

Package delivery service of a shipper is considered. This chapter considers two delivery modes, i.e., truck delivery mode and drone delivery mode (Figure 3.1). The truck delivery mode is for the shipper to rent vehicles, e.g., lorry trucks, and use the vehicles to deliver packages to customers. The drone delivery mode is for the shipper to rent drones and use the drones to deliver packages to customers. While trucks have a higher initial cost, e.g., rental fee and driver wage, and a higher traveling cost, i.e., fuel, than those of the drones, the trucks have larger capacity and are more reliable.

Drone delivery is less reliable than the truck delivery as the drones can be affected by weather and other technical problems. Consequently, the drone delivery has

²In this chapter, total cost and total payment refer to the same.

FIGURE 3.1: The joint ground and aerial delivery service optimisation and planning (GADOP) framework

FIGURE 3.2: The time-line of the scenarios.

FIGURE 3.3: An example of scenario tree with one drone.

a high breakdown probability. This chapter specifically considers two operability conditions of the drone, i.e., a takeoff condition and a breakdown condition. The takeoff condition indicates whether the drone can take off from the depot or not, which is observable before the drone departs. For example, the drone cannot takeoff from the depot if there is rain or strong wind. The breakdown condition happens after the drone already takes off from the depot, e.g., due to an accident. Therefore, the malfunctioned drone needs to be collected from the site and repaired, which also incurs a certain cost. Furthermore, the shipper will pay a penalty composed of the fee for outsourcing the delivery to a carrier and the compensation to the customer due to the delayed delivery.

The shipper faces a problem of utilising trucks and drones for package delivery for their customers. Trucks and drones need to be reserved in advance. Furthermore, the shipper has to take the uncertainties including drone takeoff and breakdown conditions into account. Therefore, the GADOP framework is formulated as a three-stage stochastic integer programming model.

- **First Stage:** the shipper decides (i) whether customers will be served by either trucks or drones, (ii) how many trucks and drones need to be reserved, (iii) the routing path of trucks, and (iv) the serving order of drones' delivery. The decisions are made based on the probability of the uncertainties.
- **Second Stage:** the shipper calculates the penalty payment for customers which cannot be served due to the drone takeoff condition.
- **Third Stage:** the shipper calculates the penalty payment for customers which cannot be served due to the drone breakdown condition. The breakdown condition is associated with each drone and each customer. Additionally, the shipper takes the repair cost into account when the breakdown happens.

Figure 3.2 shows the stages of the GADOP framework. In the first stage, the decisions are made before the actual drone takeoff condition is observed. In the second stage, the decisions are made after the takeoff condition is observed and before the delivery begins. The third stage decisions are made when an actual breakdown of the drone is observed. Figure 3.3 shows an example of a scenario

tree with one drone for the shipper. The formal definition of the scenarios is presented in Section 3.1.1.

The shipper has a set of customers to be served denoted by $\mathcal{C} = \{C_1, C_2, \dots, C_{c'}\}$, where c' denotes the total number of customers. Without loss of generality, each customer in $\mathcal{C}$ has one package to be delivered, and g_i denotes the package weight of customer i . The shipper will assign customers' packages to either trucks or drones in the first stage. I assume that all the customers have a dropoff station for the drone delivery. The shipper can choose to reserve and use trucks from the set $\mathcal{T} = \{T_1, T_2, \dots, T_{t'}\}$ and drones from the set $\mathcal{D} = \{D_1, D_2, \dots, D_{d'}\}$. t' and d' denote the total numbers of available trucks and drones, respectively. Every truck in the set $\mathcal{T}$ has a capacity limit ($\bar{f}_t$), daily distance traveling limit ($\bar{l}_t$), and daily traveling time limit ($\bar{h}_t$). The package dropoff time of a truck is denoted by $\bar{r}$. Similarly, every drone in the set $\mathcal{D}$ has a capacity limit ($\hat{f}_d$), daily flying distance limit ($\hat{l}_d$), and flying distance limit per trip ($\hat{e}_d$). The package dropoff time of a drone is assumed to be negligible.

The shipper has one depot. The traveling distance from location i' to location j' is represented as $k_{i',j'}$, where i' and j' are the locations being members of the set $\mathcal{C} \cup \{0\}$, where 0 represents the depot. The average truck driving speed and the drone flying speed are denoted by $\bar{q}_{i',j'}$ and $\hat{q}_{i',j'}$, respectively. I assume that a drone can carry only one package per trip, and therefore the drone needs to return to the depot after delivering one package. A similar assumption is made in the literature, e.g., [11], [74].

3.1.1 Uncertainty

The drone delivery is much less reliable than the truck delivery [32]. Drones are directly influenced by many unpredictable cases before taking off from their depot such as weather, mechanical issue, and organisation policy. The clarification of each factors is listed as follows.

- Weather: drones cannot take off when the wind is too strong or when raining.
- mechanical issue: drones cannot take off due to mechanical issue, e.g. (i) improper setting, (ii) battery is not sufficient due to unprepared, (iii) loss of

signal due to a telecommunication provider problem, and (iv) drone hardware problem.

- Organisation policy: the delivery organiser may directly influence to drones take off. For example, a drone is ready to take off today, without no weather issue and technical issue found, but the drone cannot pick up the assigned package because the package is not ready, e.g. (i) the package is not in the proper condition for the drone to carry, (ii) the drone is unable to find the assigned package in the depot, (iii) the information of the package is incorrect.

Therefore, the uncertainty of the drones including takeoff and breakdown conditions is considered. Again, the takeoff condition and breakdown condition are observable in the second and third stages, i.e., before and after taking off, respectively.

Let $\Omega = \{\omega_1, \omega_2, \dots, \omega_{\omega'}\}$ be the set of the takeoff scenarios, where ω' represents the total number of the scenarios. The takeoff scenario is defined as $\omega = (\mathbb{R}_1, \mathbb{R}_2, \dots, \mathbb{R}_{d'})$, the maximum number of takeoff scenarios is based on the total number of drones, i.e., d' . $\mathbb{R}_d = 1$ when drone D_d cannot take off from the depot, and $\mathbb{R}_d = 0$ otherwise. For example, $\omega = (0, 0, 1)$ means that the shipper has three drones, and drones D_1 and D_2 can take off while drone D_3 cannot take off from the depot. Let $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_{\lambda'}\}$ denote the set of breakdown scenarios, where $\lambda_{\lambda'}$ represents the total number of the scenario. The breakdown scenario λ contains the breakdown parameters indicating which drones are broken during the delivery. The scenario is defined as follows:

$$\lambda = \begin{bmatrix} \mathbb{B}_{1,1} & \mathbb{B}_{1,2} & \dots & \mathbb{B}_{1,d'} \\ \mathbb{B}_{2,1} & \mathbb{B}_{2,2} & \dots & \mathbb{B}_{2,d'} \\ \vdots & \vdots & \dots & \vdots \\ \mathbb{B}_{c',1} & \mathbb{B}_{c',2} & \dots & \mathbb{B}_{c',d'} \end{bmatrix}, \quad (3.1)$$

where $\mathbb{B}_{i,d} = 1$ when drone D_d breaks down while serving customer C_i , and $\mathbb{B}_{i,d} = 0$ otherwise. Additionally, $\mathbb{P}(\omega)$ and $\mathbb{P}(\lambda)$ denote the probabilities that scenario ω and scenario λ will happen, respectively. In reality, the probabilities can be obtained from the history records.

3.1.2 Payments

The shipper involves six types of payments, i.e., costs, including (i) the initial costs of trucks, (ii) the initial costs of drones, (iii) the traveling costs of trucks, (iv) the traveling costs of drones, (v) penalty of unsuccessful drone delivery, and (vi) drone repair cost. The initial costs are the fixed costs such as rental fee, driver or staff wages. $\bar{c}_t^{(i)}$ and $\hat{c}_d^{(i)}$ represent the initial costs of trucks and the initial costs of drones, respectively. The traveling costs are functions of traveling distance, fuel cost, or battery recharge cost. $\bar{c}_t^{(r)}$ and $\hat{c}_d^{(r)}$ represent the traveling costs of trucks and the traveling costs of drones, respectively. The penalty and the repair cost, denoted as p and m , respectively, are incurred when a drone does not successfully deliver a package.

The initial costs of trucks and drones are highly important parameters to the GADOP framework since these parameters will control how many trucks or drones should be used for the delivery. For example, if the initial costs of trucks are very high, the framework will assign more customers to drones, and vice versa.

In general, the traveling costs of trucks and drones are very low when comparing to the other costs. Thus, the traveling costs have not significant impact to the framework.

The penalty of unsuccessful drone delivery and the repair cost directly depend on the scenarios and the probability of each scenario. However, if the penalty is too high, the GADOP framework will assign more customers to the trucks as it does not want to pay for the penalty, which is similar to the repair cost.

3.2 GADOP problem formulation: single objective

$$\text{Minimise:} \quad \sum_{t \in \mathcal{T}} \bar{c}_t^{(i)} \bar{W}_t + \sum_{d \in \mathcal{D}} \hat{c}_d^{(i)} \widehat{W}_d + \sum_{\substack{i', j' \in \\ \mathcal{C} \cup \{0\}}} \sum_{t \in \mathcal{T}} \bar{c}_{i', j'}^{(r)} V_{i', j', t} + \mathbb{E}(\mathcal{L}(\widehat{X}_{i, d})), \quad (3.2)$$

where

$$\mathcal{L}(\hat{X}_{i,d}) = \sum_{\omega \in \Omega} \mathbb{P}(\omega) \left(\sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} \hat{c}_i^{(r)} \hat{X}_{i,d} (1 - \mathbb{R}_d(\omega)) + \sum_{d \in \mathcal{D}} p Z_{i,d}^{(b)}(\omega) + \mathbb{E}(\mathcal{M}(\hat{X}_{i,d}, \omega)) \right), \quad (3.3)$$

$$\mathcal{M}(\hat{X}_{i,d}, \omega) = \sum_{\lambda \in \Lambda} \mathbb{P}(\lambda) \left(\sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} p Z_{i,d}^{(a)}(\omega, \lambda) + \sum_{d \in \mathcal{D}} m Z_d^{(m)}(\omega, \lambda) \right), \quad (3.4)$$

subject to: (3.9) to (3.8).

The GADOP framework is a three-stage stochastic integer programming as modelled in (3.2) for the first stage, (3.3) for the second stage, and (3.4) for the third stage. $\mathbb{E}(\mathcal{L}(\hat{X}_{i,d}))$ denotes the expectation of the traveling cost and the penalty of drones over random takeoff conditions occurred in the second stage. $\mathbb{E}(\mathcal{M}(\hat{X}_{i,d}, \omega))$ denotes the expectation of the penalty cost and the repair cost of drones over random takeoff conditions and random breakdown conditions occurred in the third stage. The objective is to minimise the total delivery payments as discussed in Section 3.1.2. The decision variables in the framework are listed below.

- $\bar{W}_t$ is an indicator whether truck t will be used or not. When $\bar{W}_t = 1$, truck t will be used, and $\bar{W}_t = 0$ otherwise.
- $\widehat{W}_d$ is an indicator whether drone d will be used or not. When $\widehat{W}_d = 1$, drone d will be used, and $\widehat{W}_d = 0$ otherwise.
- $V_{i',j',t}$ is a routing variable in which $V_{i',j',t} = 1$ if the route from location i' to location j' will be used by truck t , and $V_{i',j',t} = 0$ otherwise.
- $\hat{X}_{i,d}$ is an allocation variable of drone d . $\hat{X}_{i,d} = 1$ if drone d will serve customer i , and $\hat{X}_{i,d} = 0$ otherwise.
- $\bar{X}_{i,t}$ is an allocation variable of truck t . $\bar{X}_{i,t} = 1$ if truck t will serve customer i , and $\bar{X}_{i,t} = 0$ otherwise.
- $Z_{i,d}^{(b)}(\omega)$ is a penalty variable. $Z_{i,d}^{(b)}(\omega) = 1$ if the penalty associated to customer i , which will be served by drone d , but the drone cannot take off from the depot, is paid in scenario ω , and $Z_{i,d}^{(b)}(\omega) = 0$ otherwise.

- $Z_{i,d}^{(a)}(\omega, \lambda)$ is another penalty variable. $Z_{i,d}^{(a)}(\omega, \lambda) = 1$ if the penalty associated to customer i , which will be served by drone d , but the drone breaks down, is paid in scenarios ω and λ , $Z_{i,d}^{(a)}(\omega, \lambda) = 0$ otherwise.
- $Z_d^{(m)}(\omega, \lambda)$ is a repair variable. $Z_d^{(m)}(\omega, \lambda) = 1$ if drone d in scenario ω and λ breaks down during delivery, and $Z_d^{(m)}(\omega, \lambda) = 0$ otherwise.
- $S_{i,t}$ is an auxiliary variable for sub-tour elimination in truck delivery.
- $U_{i,d}$ is a serving order of the drones. $U_{i,d} < U_{j,d}$ means that drone d will serve customer i before customer j .
- $M_{i,j,d}$ is an auxiliary binary variable for ensuring the nonequivalent of $U_{i,d}$.

The types and bounds of all decision variables are indicated in constraints (3.5) to (3.8).

$$\widehat{X}_{i,d}, \bar{X}_{i,t}, \widehat{W}_d, \bar{W}_t \in \{0, 1\}, \quad \forall d \in \mathcal{D}, \forall t \in \mathcal{T}, \forall i \in \mathcal{C} \quad (3.5)$$

$$V_{i',j',t} \in \{0, 1\}, \quad \forall i', j' \in \mathcal{C} \cup \{0\}, \forall t \in \mathcal{T} \quad (3.6)$$

$$U_{i,d}, S_{i,t} \in \{0, 1, \dots, c'\}, \quad \forall d \in \mathcal{D}, \forall t \in \mathcal{T}, \forall i \in \mathcal{C} \quad (3.7)$$

$$Z_{i,d}^{(b)}(\omega), Z_{i,d}^{(a)}(\omega, \lambda), Z_d^{(m)}(\omega, \lambda), M_{i,j,d} \in \{0, 1\}, \quad \forall d \in \mathcal{D}, \forall i, j \in \mathcal{C}, \omega \in \Omega, \lambda \in \Lambda \quad (3.8)$$

The constraints in (3.9) and (3.10) ensure that the initial cost of trucks and the initial cost of drones will be paid if they are used, where Δ is any large number $\Delta \geq c'$. The constraints in (3.11) and (3.12) ensure that the total weight of packages does not exceed the capacity limit of a truck and a drone, respectively. g_i denotes the weight of package that belongs to customer i . The constraint in (3.13) ensures that the flying distance between the depot to a customer and vice versa does not exceed the flying limit. The daily traveling distance limits are controlled by the constraints in (3.14) and (3.15) for drone delivery and truck delivery, respectively. The total time of the truck delivery, i.e., traveling time and serving time, must not exceed the working hours as imposed in the constraint in (3.16). Again, the traveling distance from location i' to location j' is represented as $k_{i',j'}$, where i' and j' are the locations being members of the set $\mathcal{C} \cup \{0\}$, where 0 represents the

depot. The average truck driving speed and the drone flying speed are denoted by $\bar{q}_{i',j'}$ and $\hat{q}_{i',j'}$, respectively.

$$\sum_{i \in \mathcal{C}} \bar{X}_{i,t} \leq \Delta \bar{W}_t, \quad \forall t \in \mathcal{T} \quad (3.9)$$

$$\sum_{i \in \mathcal{C}} \hat{X}_{i,d} \leq \Delta \hat{W}_d, \quad \forall d \in \mathcal{D} \quad (3.10)$$

$$\sum_{i \in \mathcal{C}} g_i \bar{X}_{i,t} \leq \bar{f}_t, \quad \forall t \in \mathcal{T} \quad (3.11)$$

$$g_i \hat{X}_{i,d} \leq \hat{f}_d, \quad \forall i \in \mathcal{C}, \forall d \in \mathcal{D} \quad (3.12)$$

$$(k_{0,i} + k_{i,0}) \hat{X}_{i,d} \leq \hat{e}_d, \quad \forall i \in \mathcal{C}, \forall d \in \mathcal{D} \quad (3.13)$$

$$\sum_{i \in \mathcal{C}} (k_{0,i} + k_{i,0}) \hat{X}_{i,d} \leq \hat{l}_d, \quad \forall d \in \mathcal{D} \quad (3.14)$$

$$\sum_{\substack{i',j' \in \\ \mathcal{C} \cup \{0\}}} k_{i',j'} V_{i',j',t} \leq \bar{l}_t, \quad \forall t \in \mathcal{T} \quad (3.15)$$

$$\sum_{\substack{i',j' \in \\ \mathcal{C} \cup \{0\}}} \left(\frac{k_{i',j'}}{q_{i',j'}} + \bar{r} \right) V_{i',j',t} \leq \bar{h}_t, \quad \forall t \in \mathcal{T} \quad (3.16)$$

The allocation of trucks and drones is imposed by the constraint in (3.17), where one customer can be served by either a truck or a drone. The constraints in (3.18) to (3.23) are to find the route for truck delivery. Only the allocated truck can take the arrival and departing route to serve the customer. The constraint in (3.23)

ensures that no subtour exists in the solution.

$$\sum_{t \in \mathcal{T}} \bar{X}_{i,t} + \sum_{d \in \mathcal{D}} \hat{X}_{i,d} = 1, \quad \forall i \in \mathcal{C} \quad (3.17)$$

$$\sum_{i \in \mathcal{C}} V_{0,i,t} \leq 1, \quad \forall t \in \mathcal{T} \quad (3.18)$$

$$\sum_{i \in \mathcal{C}} V_{i,0,t} \leq 1, \quad \forall t \in \mathcal{T} \quad (3.19)$$

$$\sum_{i' \in \mathcal{C} \cup \{0\}} V_{i',i,t} = \bar{X}_{i,t}, \quad \forall t \in \mathcal{T}, \forall i \in \mathcal{C} \quad (3.20)$$

$$\sum_{i' \in \mathcal{C} \cup \{0\}} V_{i,i',t} = \bar{X}_{i,t}, \quad \forall t \in \mathcal{T}, \forall i \in \mathcal{C} \quad (3.21)$$

$$V_{i',i',t} = 0, \quad \forall t \in \mathcal{T}, \forall i' \in \mathcal{C} \cup \{0\} \quad (3.22)$$

$$S_{i,t} - S_{j,t} + c' V_{i,j,t} \leq c' - 1, \quad \forall i \in \mathcal{C}, \forall j \in \mathcal{C}, \forall t \in \mathcal{T} \quad (3.23)$$

$$\hat{X}_{i,d} \mathbb{R}_d(\omega) = Z_{i,d}^{(b)}(\omega), \quad \forall i \in \mathcal{C}, \forall d \in \mathcal{D}, \forall \omega \in \Omega \quad (3.24)$$

$$\hat{X}_{i,d} (1 - \mathbb{R}_d(\omega)) \mathbb{B}_{i,d}(\lambda) \leq Z_{i,d}^{(a)}(\omega, \lambda), \quad \forall i \in \mathcal{C}, \forall d \in \mathcal{D}, \forall \omega \in \Omega, \lambda \in \Lambda \quad (3.25)$$

If drones cannot take off from the depot, the penalty must be paid for all associated customers, as indicated in the constraint in (3.24). In the case that the drones can take off, but they are broken after departing, the constraint in (3.25) ensures that the penalty must be paid. When breakdown occurs after departing from the depot, the drone will be no longer able to serve all the next customers. Consequently, the penalty associated with these customers must be paid as indicated in the constraint in (3.26). Note that the next customer j after customer i has a higher serving order, i.e., $U_{j,d} > U_{i,d}$. Moreover, the constraint in (3.27) ensures that the repair cost must be paid if the drone is broken after departing from the depot.

$$(U_{i,d} - U_{j,d}) \leq \Delta \left(1 - Z_{j,d}^{(a)}(\omega, \lambda) + Z_{i,d}^{(a)}(\omega, \lambda) \right), \quad \forall i, j \in \mathcal{C}, i \neq j, \forall d \in \mathcal{D}, \forall \omega \in \Omega, \lambda \in \Lambda \quad (3.26)$$

$$Z_d^{(m)}(\omega, \lambda) \geq Z_{i,d}^{(a)}(\omega, \lambda), \quad \forall i \in \mathcal{C}, \forall d \in \mathcal{D}, \forall \omega \in \Omega, \lambda \in \Lambda \quad (3.27)$$

$$\hat{X}_{i,d} \leq U_{i,d}, \quad \forall d \in \mathcal{D}, \forall i \in \mathcal{C} \quad (3.28)$$

$$0 \leq U_{j,d} \leq \sum_{i \in \mathcal{C}} \hat{X}_{i,d}, \quad \forall d \in \mathcal{D}, \forall j \in \mathcal{C} \quad (3.29)$$

$$U_{i,d} - U_{j,d} \leq \Delta M_{i,j,d} - \hat{X}_{i,d}, \quad \forall i, j \in \mathcal{C}, i \neq j, \forall d \in \mathcal{D} \quad (3.30)$$

$$U_{i,d} - U_{j,d} \geq \hat{X}_{i,d} - \Delta(1 - M_{i,j,d}), \quad \forall i, j \in \mathcal{C}, i \neq j, \forall d \in \mathcal{D} \quad (3.31)$$

The constraints in (3.28) to (3.31) decide serving orders of drones. The constraint in (3.28) ensures that $U_{i,d}$ must not be zero when drone d is allocated to serve customer i , i.e., $\ddot{X}_{i,d} = 1$. $U_{i,d} = 0$ which represents the drone serving order is thus not needed. The constraints in (3.30) and (3.31) ensure that the serving orders must not be the same when $\ddot{X}_{i,d} = 1$. Again, $M_{i,j,d}$ is an auxiliary binary variable for ensuring that $U_{i,d} \neq U_{j,d}$ when $\hat{X}_{i,d} = 1$.

In order to enhance customers' satisfaction, shippers allow their customers to specific a delivery time slot, which can be referred to as time windows. Time windows constraint can be hard time windows or soft time windows. In the hard time window, all customers must be served within time windows. On the other hand, the soft time window is more relaxed, and thus the shipper can choose to pay for the penalty of not delivery on time. This chapter considers the hard time window. The constraints in (3.32) and (3.33) ensures that the delivery is done within the assigned time windows. $\mathcal{L}^{(m)}$ and $\mathcal{L}^{(a)}$ denote a set of customers who require to be served in the morning and afternoon, respectively, and I have $\mathcal{L}^{(m)} \subseteq \mathcal{C}$, $\mathcal{L}^{(a)} \subseteq \mathcal{C}$ and $\mathcal{L}^{(m)} \cap \mathcal{L}^{(a)} = \emptyset$.

The constraints in (3.32) and (3.33) ensure that all morning customers must be served before the afternoon customers. The constraints in (3.34) to (3.37) ensure that traveling time of trucks and drones does not exceed the morning and afternoon traveling time limits, i.e., time windows. The morning and afternoon traveling time limits are defined as $l^{(m)}$ and $l^{(a)}$, respectively. More specific time slots and explanation of hard time window can be found in [127].

$$U_{i,d} \leq U_{j,d} + \Delta(1 - \hat{X}_{i,t}), \quad \forall i \in \mathcal{L}^{(m)}, \forall j \in \mathcal{L}^{(a)}, \forall d \in \mathcal{D}, \forall \omega \in \Omega \quad (3.32)$$

$$S_{i,t} \leq S_{j,t} + \Delta(1 - \bar{X}_{i,t}), \quad \forall i \in \mathcal{L}^{(m)}, \forall j \in \mathcal{L}^{(a)}, \forall t \in \mathcal{T}, \forall \omega \in \Omega \quad (3.33)$$

$$\sum_{m \in \mathcal{L}^{(m)}} \left(\frac{k_{0,m}}{\widehat{q}_{0,m}} + \frac{k_{m,0}}{\widehat{q}_{m,0}} \right) \widehat{X}_{m,d} \leq l^{(m)}, \quad \forall d \in \mathcal{D} \quad (3.34)$$

$$\sum_{a \in \mathcal{L}^{(a)}} \left(\frac{k_{0,a}}{\widehat{q}_{0,a}} + \frac{k_{a,0}}{\widehat{q}_{a,0}} \right) \widehat{X}_{a,d} \leq l^{(a)}, \quad \forall d \in \mathcal{D} \quad (3.35)$$

$$\sum_{i' \in \mathcal{C} \cup \{0\}} \sum_{m \in \mathcal{L}^{(m)}} \left(\frac{k_{i',m}}{\widehat{q}_{i',m}} + \bar{r} \right) V_{i',m,t} \leq l^{(m)}, \quad \forall t \in \mathcal{T} \quad (3.36)$$

$$\sum_{i \in \mathcal{C}} \sum_{a' \in \mathcal{L}^{(a)} \cup \{0\}} \left(\frac{k_{i,a'}}{\widehat{q}_{i,a'}} + \bar{r} \right) V_{i,a',t} \leq l^{(a)}, \quad \forall t \in \mathcal{T} \quad (3.37)$$

The above optimisation problem can be complex to solve when the numbers of customers, trucks, and drones grow. Therefore, the next section presents the decomposition method to obtain an optimal solution of the problem more efficiently.

3.3 Solution technique: decomposition

From the optimisation formulations of VRP and the GADOP, I can simply find the GADOP problem is a generalisation of the traditional VRP. For example, when the initial cost of drones is extremely high, the GADOP will assign packages to trucks. When the drones are not being used, the GADOP is reduced and becomes the VRP. Since VRP is known to be an NP-hard problem, the generalisation of the VRP is also an NP-hard, and thus the GADOP problem is an NP-hard.

The computational time for solving an NP-hard problem depends on the numbers of variables, parameters, and constraints. To address the computational issue, I decompose the GADOP optimisation into multiple sub-problems according to the decision stages, i.e., one first-stage problem, ω' second-stage sub-problems, and ω' third-stage sub-problems. Each second-stage sub-problem and each third-stage sub-problem are associated with a certain takeoff scenario, i.e., $\omega \in \Omega$. The decomposition is executed iteratively as described in the following steps:

Step 1 Solving the first-stage problem, which is the master problem,

Step 2 Solving all the second-stage sub-problems and the third-stage sub-problems using the results from the master problem as the parameters,

FIGURE 3.4: Decomposition of scenarios and stages where a circle represents one optimisation master problem or sub-problem.

Step 3 Calculating feedback parameters, i.e., $E_{i,d}^{(p)}$, $E_{i,d}^{(r)}$, and $E_{i,d}^{(m)}$ representing the penalty, the traveling cost of drones, and drone repair cost from the second-stage sub-problems and third-stage sub-problem, and

Step 4 Checking the convergence condition, if the convergence condition is met, then the solution is found, and otherwise the feedback parameters are sent to the master problem, and Step (i) repeats with the additional constraints applying the feedback parameters.

This decomposition is based on the L-Shape method [64]. The algorithm to execute the decomposition is shown in Algorithm 1.

Figure 3.4 presents an example of the GADOP optimisation with five takeoff scenarios and ten breakdown scenarios. After decomposition, the example has one master problem, five second-stage sub-problems, and five third-stage sub-problems to be solved. Note that the number of the second-stage sub-problems and the third-stage sub-problems depend on the number of the takeoff scenarios. In fact, after applying the decomposition, I have found that the solution of the second-stage problem can be obtained by a simple calculation due to the special structure of the GADOP problem. As a result, for the same takeoff scenario, the second-stage sub-problem and the third-stage sub-problem do not have any interdependency with each other, and thus I can solve the second-stage sub-problems and the third-stage sub-problems simultaneously. In the GADOP framework, I assume that a takeoff scenario does not have any interdependency with other takeoff scenario. However,

dependencies of breakdown scenarios exist in serving orders $(U_{i,d})$. Since the serving orders need to be decided before the delivery begins, the same serving order $(U_{i,d})$ is used for all breakdown scenarios. As a result, the third-stage optimisation cannot be further decomposed according to breakdown scenarios.

When the GADOP framework considers a problem with 40 customers, two drones, four takeoff scenarios, and 10 breakdown scenarios. The sub-optimisation problems are listed as follows.

- one master problem, i.e., the first stage problem.
- four second-stage sub-problems based on the number of takeoff scenarios.
- forty third-stage sub-problems based on the number of takeoff scenarios multiplied by the number of the breakdown scenarios.

The total number of sub-problems is 45 (optimisation problem). The number of customers does not affect the number of decomposed sub-problems. Note that the GADOP framework has no upper limit of the number of customers. However, when the number of customers grows, the computational time will be slower. Note that my studies do not focus on the solving algorithm, e.g., heuristic.

3.3.1 Master problem or first-stage problem

Minimising:

$$\sum_{t \in \mathcal{T}} \bar{c}_t^{(i)} \bar{W}_t + \sum_{d \in \mathcal{D}} \bar{c}_d^{(i)} \widehat{W}_d + \sum_{\substack{i', j' \in \\ \mathcal{C} \cup \{0\}}} \sum_{t \in \mathcal{T}} \bar{c}_{i', j'}^{(r)} V_{i', j', t} + \theta_1 + \theta_2, \quad (3.38)$$

subject to: (3.9) to (3.23), (3.33) to (3.37), (3.39), and (3.40).

$$\sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} (E_{i,d}^{(p)} + E_{i,d}^{(r)}) \widehat{X}_{i,d} + \theta_1 \geq 0 \quad (3.39)$$

$$\sum_{d \in \mathcal{D}} E_{i,d}^{(m)} \widehat{W}_d + \theta_2 \geq 0, \quad \forall i \in \mathcal{C} \quad (3.40)$$

The objective function of the master problem is presented in (3.38). The constraints in (3.39) and (3.40) are the feedback cutting constraints [64]. Again, $E_{i,d}^{(p)}$, $E_{i,d}^{(r)}$,

and $E_{i,d}^{(m)}$ are the feedback parameters, which are dedicated to the penalty, the drone traveling cost, and the drone repair cost, respectively. θ_1 and θ_2 are initially set at small values for the first iteration. After the first iteration, θ_1 and θ_2 become positive decision variables. The results of θ_1 and θ_2 are denoted by θ_1^* and θ_2^* , respectively, which are used in checking the convergence condition. The result of $\hat{X}_{i,d}$ is denoted by $\hat{X}_{i,d}^*$, which is used as a parameter in the second-stage sub-problems and the third-stage sub-problems. After decomposition, the first-stage problem is still an NP-hard problem. However, the first-stage problem becomes less complex than the original GADOP because the number of decision variables in the first-stage problem is fewer than that in the original problem.

3.3.2 Second-stage sub-problem

Minimise:

$$\sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} \hat{c}_i^{(r)} \hat{X}_{i,d}^* (1 - \tilde{\mathbb{R}}_d) + \sum_{d \in \mathcal{D}} p \tilde{Z}_{i,d}^{(b)}, \quad (3.41)$$

subject to: (3.42).

$$\hat{X}_{i,d}^* \tilde{\mathbb{R}}_d = \tilde{Z}_{i,d}^{(b)}, \quad \forall i \in \mathcal{C}, \forall d \in \mathcal{D} \quad (3.42)$$

I decompose the second-stage sub-problem given a certain takeoff scenario, and thus the variable ω is removed from the objective function in (3.41) and the constraint in (3.42). To minimise the expected cost in the second stage, i.e., $\mathbb{E}(\mathcal{L}(\hat{X}_{i,u}))$, I need to solve all the second-stage sub-problems associated with a takeoff scenario (ω). Note that random parameters $\tilde{\mathbb{R}}_d$ and $\tilde{\mathbb{B}}_{i,d}$, i.e., for scenario variables $\mathbb{R}_d$ and $\mathbb{B}_{i,d}$ defined in Section 3.1.1, respectively, as well as the decision variable $\tilde{Z}_{i,d}^{(b)}$ is associated with the takeoff scenario, and therefore the ω is omitted.

In general, the second-stage problem is an optimisation, and thus I present the second-stage problem in the optimisation form. Nevertheless, in my exact problem, there is a special structure that makes the second-stage problem is able to be solved directly through the assignment of $\hat{X}_{i,d}$ and $\tilde{\mathbb{R}}_d$.

Before applying the decomposition, the decision variables, which are involved in the second-stage, are variables $\hat{X}_{i,d}$ and $\tilde{Z}_{i,d}^{(b)}$. However, after applying the decomposition, there is only one decision variable in the second-stage problem, i.e., $\tilde{Z}_{i,d}^{(b)}$.

The reason is that in the decomposed version, the decision variable $\hat{X}_{i,d}$ is solved in the first-stage problem. Thus, variable $\hat{X}_{i,d}$ becomes a parameter, i.e., $\hat{X}_{i,d}^*$, in the second-stage problem. According to the constraint in (3.42), variable $\tilde{Z}_{i,d}^{(b)}$ depends on parameters only, and consequently the second-stage problem can be simply solved from the constraint in (3.42).

3.3.3 Third-stage sub-problem

Minimise:

$$\sum_{\lambda \in \Lambda} \mathbb{P}(\lambda) \left(\sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} p \tilde{Z}_{i,d}^{(a)}(\lambda) + \sum_{d \in \mathcal{D}} m \tilde{Z}_d^{(m)}(\lambda) \right), \quad (3.43)$$

subject to: (3.28), (3.44) to (3.50).

$$\hat{X}_{i,d}^* (1 - \tilde{\mathbb{R}}_d) \tilde{\mathbb{B}}_{i,d} \leq \tilde{Z}_{i,d}^{(a)}(\lambda), \quad \forall i \in \mathcal{C}, \forall d \in \mathcal{D} \quad (3.44)$$

$$U_{i,d} - U_{j,d} \leq \Delta \left(1 - \tilde{Z}_{j,d}^{(a)}(\lambda) + \tilde{Z}_{i,d}^{(a)}(\lambda) \right), \quad \forall i, j \in \mathcal{C}, i \neq j, \forall d \in \mathcal{D} \quad (3.45)$$

$$\tilde{Z}_d^{(m)}(\lambda) \geq \tilde{Z}_{i,d}^{(a)}(\lambda), \quad \forall i \in \mathcal{C}, \forall d \in \mathcal{D} \quad (3.46)$$

$$\hat{X}_{i,d}^* \leq U_{i,d}, \quad \forall d \in \mathcal{D}, \forall i \in \mathcal{C} \quad (3.47)$$

$$0 \leq U_{j,d} \leq \sum_{i \in \mathcal{C}} \hat{X}_{i,d}^*, \quad \forall d \in \mathcal{D}, \forall j \in \mathcal{C} \quad (3.48)$$

$$U_{i,d} - U_{j,d} \leq \Delta M_{i,j,d} - \hat{X}_{i,d}^*, \quad \forall i, j \in \mathcal{C}, i \neq j, \forall d \in \mathcal{D} \quad (3.49)$$

$$U_{i,d} - U_{j,d} \geq \hat{X}_{i,d}^* - \Delta(1 - M_{i,j,d}), \quad \forall i, j \in \mathcal{C}, i \neq j, \forall d \in \mathcal{D} \quad (3.50)$$

Similar to the second-stage sub-problem, each third-stage sub-problem is formulated given a certain takeoff scenario. The objective function of the third-stage sub-problem is expressed in (3.43) which is the expected cost of the penalty and repair cost of drones. The constraints in (3.44), (3.45), and (3.46) contain the third-stage decision variables, which are $\tilde{Z}_{i,d}^{(a)}(\lambda)$ and $\tilde{Z}_d^{(m)}(\lambda)$. The decision variables $\tilde{Z}_{i,d}^{(a)}(\lambda)$ and $\tilde{Z}_d^{(m)}(\lambda)$ are associated with the takeoff scenario, and therefore ω is omitted. Since there is the interdependency among $U_{i,d}$, i.e., drone serving order, the constraints in (3.47)-(3.50) are directly adopted in this sub-problem. The reason is that $U_{i,d}$ of each takeoff scenario, which depends on $\hat{X}_{i,d}^*$, has already been obtained in the master problem. Again, $\hat{X}_{i,d}^*$ is the assignment of a drone to a customer.

3.3.4 Calculating feedback parameters and convergence

This section explains the feedback parameter calculation and the convergence condition checking. $e_{i,d}^{(p)}(\omega)$ and $e_{i,d}^{(m)}(\omega)$ are the auxiliary parameters in the third-stage sub-problems, which are associated with the drone penalty and the drone repair cost, respectively. The feedback parameters, i.e., $E_{i,d}^{(r)}$, $E_{i,d}^{(p)}$, and $E_{i,d}^{(m)}$, as expressed in (3.51), (3.52), and (3.54), respectively, are returned to the master problem. $E_{i,d}^{(r)}$, $E_{i,d}^{(p)}$, and $E_{i,d}^{(m)}$ are the expected values associated with the drone traveling cost, the drone penalty, and the drone repair cost, respectively. The constraints in (3.39) and (3.40) ensure that the shipper pays the operational cost, the penalty, and the repair cost from the second-stage and third-stage when the associated $\hat{X}_{i,d}$ is selected. Therefore, $E_{i,d}^{(r)}$, $E_{i,d}^{(p)}$, and $E_{i,d}^{(m)}$ must be negative in order to match with θ_1 and θ_2 . The formulations of the feedback parameters are given as follow.

$$E_{i,d}^{(r)} = -\hat{c}_{i,d}^{(r)} \sum_{\omega \in \Omega} \mathbb{P}(\omega) (1 - \mathbb{R}_d(\omega)), \quad (3.51)$$

$$E_{i,d}^{(p)} = -\sum_{\omega \in \Omega} \mathbb{P}(\omega) \left(p \mathbb{R}_d(\omega) + e_{i,d}^{(p)}(\omega) \right), \quad (3.52)$$

$$\text{where } e_{i,d}^{(p)}(\omega) = p \sum_{\lambda \in \Lambda} \mathbb{P}(\lambda) (1 - \mathbb{R}_d(\omega)) \mathbb{B}_{i,d}(\lambda), \quad (3.53)$$

$$E_{i,d}^{(m)} = -\sum_{\omega \in \Omega} \mathbb{P}(\omega) e_{i,d}^{(m)}(\omega), \quad (3.54)$$

$$\text{where } e_{i,d}^{(m)}(\omega) = m \sum_{\lambda \in \Lambda} \mathbb{P}(\lambda) (1 - \mathbb{R}_d(\omega)) \mathbb{B}_{i,d}(\lambda). \quad (3.55)$$

I next define a convergence parameter denoted by B which is used for checking the convergence condition. If B is less than or equal to $\theta_1^* + \theta_2^*$, the convergence condition is met, and then Algorithm 1 is terminated and the solution is taken from the latest iteration. The convergence parameter is calculated from (3.56) and (3.57).

The proof of convergence of the L-shape method can be found in [64]. In the GADOP optimisation problem, I observe that the feedback constraints are based on the probabilities of scenarios, the penalty, the operational cost, and the repair cost. As I observe from the GADOP problem, these feedback parameters can be calculated directly from parameters, i.e., the probabilities of scenarios ($\mathbb{P}(\omega), \mathbb{P}(\lambda)$), the penalty (p), the operational cost ($\hat{C}_{i,d}^{(r)}$), and the repair cost (m). Since the feedback parameters do not depend on the solution at an iteration k , the feedback

Algorithm 1: Decomposition algorithms, where k is the iteration index.

Result: The optimal solution is taken from the latest iteration

while *Convergence condition is not met* **do**

if $k = 0$ **then**

 //first iteration;

 - θ_1^{*k} and θ_2^{*k} are set at very small values;

 - Solving master problem without (3.39) and (3.40);

else

 - Generating the feedback constraints as in (3.39) and (3.40) by using $E_{i,d}^{(p),k-1}$, $E_{i,d}^{(m),k-1}$, and $E_{i,d}^{(r),k-1}$;

 - Solving master problem with the feedback constraints;

end

 - Obtaining $X_{i,d}^{*k}$, θ_1^{*k} , and θ_2^{*k} ;

foreach *scenario* $\omega \in \Omega$ **do**

 - Solving the second-stage sub-problem by using $X_{i,d}^{*k}$ as parameters;

 - Solving the third-stage sub-problem by using $X_{i,d}^{*k}$ as parameters;

end

 - Calculating $E_{i,d}^{(p),k}$, $E_{i,d}^{(m),k}$, and $E_{i,d}^{(r),k}$, $\forall d \in \mathcal{D}, \forall i \in \mathcal{C}$, as in (3.52), (3.54), and (3.51), respectively;

 - Calculating B as in (3.56);

if $\theta_1^{*k} + \theta_2^{*k} \geq B$ **then**

 - Convergence condition is met;

else

 - $k \leftarrow k + 1$;

end

end

terms are constants. According to Algorithm 1, I observe that the first-stage solutions at $k = 1$ and $k = 2$ are identical because the feedback terms are the same. Therefore, the convergence of the decomposed GADOP is guaranteed at $k = 2$.

$$B = \sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} \widehat{X}_{i,d}^* (E_{i,d}^{(p)} + E_{i,d}^{(r)}) + M_d \widehat{W}_d^*, \quad (3.56)$$

$$M_d = \max(E_{i,d}^{(m)}), \quad \forall i \in \mathcal{D} \quad (3.57)$$

3.3.5 The complexity of the decomposition

From the GADOP optimization formulation, the total number of decision variables can be calculated from Equation (3.58), where t' , d' , and c' denote the total numbers of trucks, drones, and customers in the set, respectively. ω' and λ' are the total numbers of takeoff and breakdown scenarios, respectively.

$$\#decisionVariables = t' + d' + 2c'(d' + t') + c'^2(t' + d') + c'd'\omega'(1 + \lambda') + d'\omega\lambda. \quad (3.58)$$

The total numbers of the constraints in the GADOP optimization can be calculated from Equation (3.59).

$$\#constraints = 8t' + 4d' + c' + 4c'd' + 3t'c' + 3c'^2d' + 2c^2t + cd\omega(1 + 2\lambda + c). \quad (3.59)$$

After decomposing the GADOP problem into multiple sub-problem, the total numbers of decision variables of the master problem and the third-stage problem can be calculated from Equations (3.60) and (3.61), respectively.

$$\#decisionVariablesMaster = t' + d' + c'd' + 2c't' + c'^2t'. \quad (3.60)$$

$$\#decisionVariablesThird = c'd'(\lambda + c + 1) + d\lambda \quad (3.61)$$

The total numbers of the constraints in the master problem and the third-stage sub-problem in the decomposed GADOP optimization can be calculated from Equations (3.62) and (3.63), respectively.

$$\#constraintsMaster = 8t' + 4d' + c' + 3c't' + 2c'd' + 2c'^2t'. \quad (3.62)$$

$$\#constraintsThird = 5c'd' + 3c'^2d' \quad (3.63)$$

Note that the second-stage sub-problem is a general linear formulation since the second-stage sub-problem is subject to only one equivalent constraint. Thus, the complexity of the second-stage sub-problem is negligible.

For example, with 20 customers, 1 truck, 3 drones, 6 takeoff scenarios, and 20 breakdown scenarios, the original GADOP problem will have 9,687 decision variables and 26,700 constraints while the decomposed GADOP will have much smaller numbers of decision variables and constraints, i.e., (i) the master problem master problem will have 504 decision variables and 1,020 constraints, and (ii) the third-stage sub-problem will have 2,520 decision variables and 3,900 constraints. Since the complexity of linear programming directly depends on the numbers of variables and constraints, the complexity of the sub-problems in the decomposed GADOP is much simpler than the original GADOP.

3.4 Evaluation

3.4.1 Parameter setting

The GADOP framework is implemented as a three-stage integer programming by using GAMS [22]. The experiments presented in this section are solved by the CPLEX solver. I use CPLEX default setting which comes with the heuristic option. Unless otherwise stated, I consider three drones and one truck in the GADOP framework. All the three drones are of the same type, the initial costs of which are $\hat{c}_1^{(i)}, \hat{c}_2^{(i)}, \hat{c}_3^{(i)} = S\100 and the capacity limit of which are $\hat{f}_1, \hat{f}_2, \hat{f}_3 = 2$ kilograms. The daily flying distance limit and the flying distance limit per trip are $\hat{l}_d = 150$ kilometres and $\hat{e}_d = 15$ kilometres for each drone, respectively. The truck is a panel van, the initial cost of which is $\bar{c}_1^{(i)} = S\$280$ and the capacity limit of which is $\bar{f}_1 = 1,060$ kilograms. The daily distance traveling limit and daily traveling time limit are $\bar{l}_t = 200$ kilometres and $\bar{h}_t = 8$ hours, respectively. The driving speed is set at $q_{i',j'} = 50$ kilometres per hour. The initial costs of the drones and the truck are synthesised based on Singapore shipping services [111]. The traveling cost of the truck is approximated to $\bar{c}_{i',j'}^{(r)} = k_{i',j'} \times 1.05 \times 0.1$ based on Singapore government statistics [79], where 1.05 is the approximate fuel price factor and 0.1 is the average fuel consumption factor. The cost of a drone traveling round-trip between location i and the depot is $\bar{c}_i^{(r)} = 0.005 \times (k_{i,0} + k_{0,i})$, where 0 is the index of the depot.

Furthermore, all customers have the same package weight $g_i = 1$ kilogram. The serving time of the truck is $\bar{r} = 15$ minutes while the serving time of the drone is

negligible because the drone just only drops a package off. The penalty and repair cost are $p = S\$20$ and $m = S\$50$, respectively. The penalty is set based on the cost of outsourcing package delivery to the Speedpost service [128] by Singpost carrier plus additional cost for collecting the package or giving the compensation to a customer. Note that all customers have no specific time windows due to the space limit. Nevertheless, the experiment associated with time windows can be found in [127].

I evaluate the GADOP framework using two geographically data sets, i.e., Solomon Benchmark suite C101 [93] and a real trace from one of the Singapore logistics companies. Since Singapore has many high buildings, i.e., residential buildings, I use Singapore road network as the drone flying paths. The distances in kilometres between one location to another are obtained from the Google Map [44]. Solomon Benchmark suite is used in all experiments except the experiment in Section 3.4.3.

Note that if no specific parameter setting is mentioned in the following experiment sections, it means that the experiments in the sections follow the general setting in this section.

3.4.2 Tradeoff between taking and not-taking uncertainty into account

I evaluate the GADOP framework with 40 customers. The Solomon benchmark suite is tested under (i) the deterministic case where the drones are always able to takeoff and no breakdown occurs and (ii) the random case where the drones may not be able to take off and breakdown may occur.

In the deterministic case, the scenarios are set as $\Omega = \{\omega_1\}$, $\Lambda = \{\lambda_1\}$, and all $\mathbb{R}_d(\omega_1) = 0$, $\mathbb{B}_{i,d}(\lambda_1) = 0$. The customer locations and the drone flying distance limit are presented in Figure 3.5a. The solution of the optimisation in the deterministic case is to serve 17 and 23 customers by one drone and one truck, respectively. The routing path of the truck is presented in Figure 3.5b, and the customers that are served by the drone are presented in Figure 3.5c. The total payment is $S\$386.29$, which includes the initial cost of the truck $S\$280$, the initial cost of the drone $S\$100$, the routing cost of the truck $S\$5.59$, and the operational

(A) customers' locations

(B) routing path of the truck

(c) the customers served by the drone

FIGURE 3.5: Solomon benchmark example with the scenario that the drones are always able to takeoff and no breakdown occurs.

cost of the drone S\$0.70. The total traveling distances of the drone and truck are 140.27 and 53.26 kilometres, respectively.

Furthermore, in the random case, there are two scenarios for each of takeoff condition and breakdown condition ($\Omega = \{\omega_1, \omega_2\}$, $\Lambda = \{\lambda_1, \lambda_2\}$). The probabilities of two takeoff scenarios are $\mathbb{P}(\omega_1) = 0.9$ and $\mathbb{P}(\omega_2) = 0.1$, where all drones can takeoff in scenario ω_1 and cannot takeoff in scenario ω_2 . The probabilities of two random breakdown scenarios are also set to $\mathbb{P}(\lambda_1) = 0.9$ and $\mathbb{P}(\lambda_2) = 0.1$. The solution of the optimisation in the random case is to serve 14 and 26 customers by one drone and one truck, respectively. The solution is presented in Figure 3.6. The total payment is S\$414.25 including the initial cost of the truck S\$280, the initial cost of the drone S\$100, the routing cost of the truck S\$5.736, the operational cost of the drone S\$0.51, and the penalty incurred from the takeoff scenarios S\$28. The total traveling distances of the drone and the truck are 114.44 and 54.63 kilometres, respectively. Obviously, only drone delivery cannot serve all customers because some customers are outside the flying area. I observe that the solution assigns the drone to serve as many customers as possible in the deterministic case because the traveling cost of the drone is cheaper. Note that daily traveling distance limit of the drone is 150 kilometre. On the contrary, in the random case, since the drone may not be able to take off or the breakdown can occur, the solution avoids the penalty and repair cost by assigning more number of customers to the truck.

3.4.3 Routing, payment, and explanation of the real dataset

I next test the GADOP framework with the real data set from one of the Singapore logistics companies. In this experiment, I use the same takeoff and breakdown scenarios as the random case in Section 3.4.2, i.e., the two takeoff scenarios and two breakdown scenarios.

The solution of the optimisation is presented in Figure 3.7, which is to use one drone to serve 13 customers and one truck to serve 27 customers. The total payment is S\$411.76 including the initial cost of the truck S\$280, the initial of the drone S\$100, the routing cost of the truck S\$5.17, the operational cost of the drone S\$0.58, and the penalty incurred from the takeoff scenarios S\$26. The total traveling distances of the drone and the truck are 129.20 and 49.31 kilometres, respectively. I observe that the truck is used for the customers which are clustered in the same area or

(A) the routing path of the truck

(B) the customers served by the drone

FIGURE 3.6: Solomon benchmark example with two takeoff and breakdown scenarios.

located along the route. Furthermore, the drone tends to be used for the customer far away from each other such as customer 1 at the bottom right corner of the map (Figure 3.7).

3.4.4 Impacts of the number of customers and drone flying limitations

I vary the number of customers and the drone flying limit per trip. In this experiment, I test the GADOP framework with one scenario which all drones can take off and no breakdown occurs, i.e., the deterministic case. The results and the locations of the customers are presented in Figure 3.8. If the drone flying limit per trip is large, i.e., 12.5 kilometres, all the customers are assigned to a single or multiple drones without trucks. However, when the number of the customers is large, i.e., 40, the truck and drone are used together. The reason is that three drones cannot serve all 40 customers in one day. If the drone flying limit per trip is small, i.e., 5 kilometres, the truck is selected to serve all the customers when the number of the customers is less than 25. The reason is that the drone alone cannot serve the customers farther than its flying distance limit while the truck is able to serve all the customers. When the number of the customers is larger than or equal to 30, the truck alone cannot serve all the customers because of its daily traveling time limit, and thus both the truck and drone are used together when the drone flying limit per trip is less than or equal to 10 kilometres.

I conduct the performance comparison between (i) stand-alone ground-based delivery in which only ground-based trucks are used to serve customers, and (ii) joint truck and drone delivery services. As presented in Figure 3.9, the stand-alone ground-based delivery is always the most expensive delivery mode for delivering small-size packages. This result clearly shows the effectiveness of the proposed joint truck and drone delivery.

3.4.5 Impact of random takeoff scenarios

I evaluate the GADOP framework with 25 customers, two takeoff scenarios, and one breakdown scenario, i.e., $\Lambda = \{\lambda_1\}$, and all the drones do not break down, i.e., $\mathbb{B}_{i,d}(\lambda_1) = 0$. The two takeoff scenarios are (i) all the drones can take off (ω_1)

(A) Drone delivery

(B) Truck delivery

FIGURE 3.7: The real package delivery problem in Singapore. (a) and (b) present the customers who are served by the drone and the truck, respectively.

(A) the total payment

(B) customer locations and drone flying limits per trip

FIGURE 3.8: Increasing number of customers.

and (ii) all the drones cannot take off (ω_2). I vary the probability of that all the drones cannot take off, i.e., $\mathbb{P}(\omega_2)$, and $\mathbb{P}(\omega_1) = 1 - \mathbb{P}(\omega_2)$. The flying distance limit is ignored in this case for the ease of presenting the impact of the takeoff scenarios. The results are presented in Figure 3.10. The drone is used to serve all the customers when the probability is less than or equal to 0.3. Otherwise, the truck is used to serve all the customers.

FIGURE 3.9: Performance comparison between truck delivery and the proposed GADOP framework.

FIGURE 3.10: Increasing the probability of that all drones cannot take off.

FIGURE 3.11: The results of varying the number of breakdown customers and the probability of the breakdown scenarios, where the bars with an asterisk symbol on the top represents the results that all customers are served by the truck.

3.4.6 Impact of random breakdown scenarios

Next I consider 25 customers, one takeoff scenario $\Omega = \{\omega_1\}$, i.e., all the drones can take off $\mathbb{R}_d(\omega_1) = 0$, and two breakdown scenarios, i.e., $\Lambda = \{\lambda_1, \lambda_2\}$. For the two breakdown scenarios, (i) there is no breakdown, i.e., all $\mathbb{B}_{i,d}(\lambda_1) = 0$, and there is breakdown for some customers. For example, breakdown percentage equal to 20% means 5 customers have breakdown and 20 customers have no breakdown, which can be referred to $\mathbb{B}_{i,d}(\lambda_2) = 1$ for $i = 1, \dots, 5$, and $\mathbb{B}_{i,d}(\lambda_2) = 0$ for $i = 6, \dots, 25$. I vary the probability of the breakdown scenario $\mathbb{P}(\lambda_2)$ and the percentage of the breakdown customers. Note that the flying distance limit is ignored. The results are presented in Figure 3.11. From the results, the GADOP framework always selects the drone to serve the customers when the percentage of the breakdown customers is less than or equal to 20% or the probability of the breakdown scenario $\mathbb{P}(\lambda_2) \leq 0.2$.

3.4.7 Impact of penalty cost

I use the same setting and scenarios as in Section 3.4.6, but the probabilities of the breakdown scenarios are equal, i.e., $\mathbb{P}(\lambda_1) = \mathbb{P}(\lambda_2) = 0.5$. The results are presented

FIGURE 3.12: Effects of increasing the penalty.

in Figure 3.12. When the breakdown penalty cost is high, i.e., $p \geq 15$, the truck is used to serve all the customers. On the contrary, the drone is used when the penalty cost is $p < 15$. The same result is also observed when the percentage of breakdown customers is 80%, 60%, and 40% when $p \geq 20, 25$, and 35, respectively. Apparently, the number of breakdown customers has a significant impact on the solution.

FIGURE 3.13: The drone repair cost is varied.

FIGURE 3.14: The comparison between the solutions from the GADOP, the EVF, and the PDSTSP.

3.4.8 Impact of repair cost

Next, I consider two specific scenarios that (i) there is no breakdown and (ii) there is breakdown to all customers. I vary the drone repair cost and the probability of these two scenarios. Figure 3.13 shows the total payments, where (a) the penalty is ignored $p = 0$ and (b) the penalty is $p = 20$. When the probability that the breakdown occurs to all customers, i.e., $\mathbb{P}(\lambda_2) = 0$, the repair cost is not taken into account. As shown in Figure 3.13a, the drone is used to serve all the customers. On the other hand, Figure 3.13b shows that the drone is selected when the probability that the breakdown occurs to all customers is $\mathbb{P}(\lambda_2) \leq 0.25$. Otherwise, the truck is used. I observe that the impact of the drone repair cost is significantly less than the penalty p and the probability of that the breakdown occurs to all customers.

3.4.9 Simulation

I conduct a simulation and compare the performance in terms of the total payments obtained from the solutions of the GADOP optimisation, expected value formulation (EVF), the parallel drone scheduling traveling salesman problem (PDSTSP) [11].

FIGURE 3.15: The comparison between the original GADOP and the decomposition. The number of takeoff scenarios and the number of drones are varied.

I adapt the PDSTSP from minimising the delivery time to minimising total payments. The EVF and adapted PSDTSP formulations can be found in the Appendix.

Here, the initial cost of a truck to S\$400 and the daily traveling time limit is set as 10 hours. For the EVF, all the decisions are optimised only for the first stage. For the PDSTSP, the uncertainties are not considered. Clearly, the total payment obtained from solving the GADOP optimisation is significantly lower than that of the EVF and the PSDTSP. The reason is that the solution of the GADOP optimisation takes into account the takeoff and breakdown conditions explicitly which lower the chance of incurring penalty, i.e., by using trucks for more customers. On the contrary, the solution of the EVF considers only the probability of unsuccessful delivery and implicitly ignores the effect of the penalty, even though the penalty is considered. Note that when the penalty is large, e.g., S\$14 and S\$16 for the cases with 30 and 25 customers, respectively, the trucks are used for all customers which is the same for both GADOP optimisation and EVF. However, the PSDTSP decisions are not influenced by the penalty. Therefore, the total payments of GADOP and the EVF are identical, and the total payment of PSDTSP is the highest.

3.4.10 Decomposition

The decomposition of the GADOP optimisation is studied. The penalty, repair cost, and the drone initial cost are $p = 5$, $m = 5$, and $\hat{c}_d^{(i)} = 10$ respectively. This experiment considers one truck and 10 breakdown scenarios ($\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_{10}\}$). The number of drones and the number of takeoff scenarios are varied. The takeoff scenarios include all possibilities of takeoff events that can occur to all the drones. To illustrate, five drones are experimented with 32 takeoff scenarios ($\Omega = \{\omega_1, \omega_2, \dots, \omega_{32}\}$), six drones are experimented with 64 takeoff scenarios, and so on.

The comparison is shown in Figure 3.15. When the number of takeoff scenarios grows, the difference between the computational time of the original and decomposed problems becomes larger. As shown in Figure 3.15, the decomposed GADOP achieves 407%, 648%, 854%, and 1372% speedup with 32, 64, 128, and 256 scenarios, respectively. Meanwhile, the decomposed GADOP always achieves lower total payment. Note that, in some cases, the CPLEX solver cannot give an optimal solution for the original problem when the problem is too large. Thus, the CPLEX solver may return a suboptimal solution instead of the optimal solution. In this regard, the suboptimal solution achieves much higher payment than that of the decomposition while the computational time of the decomposition may not significantly improve [126]. In summary, the decomposition enables the GADOP to handle larger number of scenarios.

3.5 Conclusion

The joint ground and aerial delivery service optimisation and planning (GADOP) framework have been proposed. The uncertainty of drone delivery has been taken into account by considering takeoff and breakdown conditions. The aim of the GADOP framework is to help a shipper to make the decision either to use ground-based trucks, drones, or the combination to serve customers. The GADOP optimisation has been formulated as a three-stage stochastic integer programming model. The performance evaluation of both Solomon benchmark suite and the Singapore real road network has been conducted. The results clearly show that using drones in the package delivery can reduce the total payment of the shipper

significantly. Therefore, [Objective 1](#) and [Objective 2](#) are achieved. Moreover, simulation results have shown that the GADOP framework always achieves the lower total payment than that of the expected value formulation (EVF) and the Parallel Drone Scheduling Traveling Salesman Problem (PDSTSP). The L-shape decomposition method has adopted to enhance the solvability of the optimisation problem, and thus [Objective 3](#) is achieved.

Chapter 4

Multi-objective Optimisation for Drone and Outsource Delivery ¹

To handle the packages and parcels that cannot be delivered by drones, a shipper may outsource those packages to a carrier. The carrier charges the shipper based on the number of packages, their weights and sizes, etc. The shipper is required to evaluate the delivery plan carefully, as outsourcing the carrier is normally more expensive than using drones. To schedule the package delivery plan, the shipper has multiple objectives to fulfil. On one hand, the shipper wants to obtain a high profit. On the other hand, the shipper is required to provide reliable services and achieve a high satisfaction from customers.

To address the aforementioned challenges, I propose a multi-objective optimisation for drone delivery (MODD) framework, which aims to help the shipper schedule and plan its package delivery. According to the objectives in Section 1.2, [Objective 1](#), [Objective 2](#), and [Objective 4](#) are accomplished in this chapter. The study in this chapter revisits [Objective 1](#) in terms of different business delivery mode. A joint truck and drone delivery is presented in Chapter 3 while a joint drone delivery and outsourcing is proposed in this chapter. As the MODD framework is an extension of the GADOP framework in Chapter 3, [Objective 2](#) is automatically revisited because of the uncertainty conditions and formulations. Again, the contributions of this chapter are listed as follows. Each contribution fulfils one of the objectives.

¹The work in this chapter has been presented in [124]

Objective 1: The trade-off between using drones and outsourcing packages to a carrier is analysed as drones may not be able to reach some customers, e.g., due to flying distance limit. Moreover, the evaluation of the MODD framework is presented with the real data from a Singapore company.

Objective 2: The optimisation is formulated as a three-stage stochastic programming to handle the uncertainties of the problematic events, i.e., takeoff condition and breakdown condition.

Objective 4: Since some industries may consider delivery problem with multiple objectives in nature, the MODD framework is formulated as a three-objective optimisation. The details of the problem objectives are as follows:

- To minimise the total delivery cost.
- To minimise the percentage of unsuccessful delivered packages. The unsuccessful delivery occurs when the drone is unable to take off from the depot or the drone breaks down during the delivery.
- To maximise the reward of on-time delivery. The customers can have one or more specific preferred time slots for delivering and receiving their packages. The time slots are then associated with different rewards to be optimised by the shipper.

ε -constraint method is used to obtain the exact solutions, and the analysis of the Pareto frontiers of the MODD framework is presented. The method is summarised as four steps, which are discussed in Section 4.2.4.

The organisation of this chapter is presented as follows. Section 4.1 presents the system model and assumptions of the proposed MODD framework. Following by Section 4.2 presents system formulations, i.e., three objective functions, decision variables and constraints. At the end, the performance evaluation of the MODD framework is presented by using real customer data set of an industry in Singapore.

4.1 System model and assumptions

In this section, I describe the MODD framework, which is formulated as a three-objective optimisation. The objectives are (i) to minimise the total delivery cost,

(ii) to minimise the percentage of unsuccessful delivered packages, and (iii) to maximise the reward of on-time delivery. The MODD framework is formulated as a three-stage stochastic programming. I adopt the takeoff and breakdown condition scenarios from chapter 3. In the framework, I also consider two types of uncertainty scenarios, i.e., takeoff condition scenario and breakdown condition scenario. The first-stage and the second-stage are separated by the observation of takeoff condition, and the observation of breakdown condition divides the second-stage and the third-stage from each other. The decisions, which are made in each stage, are as follows:

- **First-stage:** Before the takeoff condition is observed, the drones are reserved, and customers are assigned to either one of the drones or outsourced to a carrier.
- **Second-stage:** The takeoff condition scenario is observed. If the drone can take off, it will deliver the package from the depot to the customer location and will return to the depot. If the drone cannot take off, all packages assigned to the drone are considered to be the unsuccessful deliveries.
- **Third-stage:** After the breakdown condition is observed, if the drone breaks down, the package in the broken drone is regarded as the unsuccessful delivery. Additionally, the packages of the customers that will be served after the breakdown occurs are also regarded as the unsuccessful deliveries.

The shipper has a set of customers to serve, which is denoted as $\mathcal{C} = \{c_1, c_2, \dots, c_{c'}\}$, where c' represents the total number of customers. I use i and j as indexes of set $\mathcal{C}$. Without loss of generality, each customer has only one package, and the weight of a package of customer i is denoted as a_i kg. The shipper can deliver customers' packages by its drones or outsource the customers' packages to a carrier. Let $\mathcal{D} = \{d_1, d_2, \dots, d_{d'}\}$ denote the set of drones, where d' is the total number of the drones. Each drone has its capacity limit (g_d), flying distance limit per trip (e_d), flying distance limit per day (l_d), start flying time ($\widehat{h}_d$), end flying time ($\bar{h}_d$), and average flying speed (s_d). If the shipper decides to serve a customer by the drone, the customer also has time preferences that he/she wants the delivery to be done in specific time slots, which can be referred to as a time window. Let $\mathcal{F} = \{f_1, f_2, \dots, f_{f'}\}$ denote the set of time windows, where f_1 and f_2 represent the most and the second-most preferred time windows, respectively, and f' represents the least preferred time window. Let $\widehat{\mathbb{T}}_i^{(f)}$ and $\bar{\mathbb{T}}_i^{(f)}$ denote the start and the end of

the time window that customer i prefers as the f^{th} order. Let t_i denote the time that a drone needs to spend while serving customer i , which can be referred to as serving time.

Moreover, the shipper has a set of depots, i.e., $\mathcal{P} = \{p_1, p_2, \dots, p_{p'}\}$, where p' represents the total number of the depots. Each drone can fly from and return to only one depot. Before delivering, customers' packages can be transferred from an original depot to a new depot, and thus a drone can take the package from the new depot instead of the original depot. Let $o_{i,p}$ be a parameter where $o_{i,p} = 1$ when the package of customer i belongs to depot p , and $o_{i,p} = 0$ otherwise. I then have the condition $\sum_{p \in \mathcal{P}} o_{i,p} = 1$ for all $i \in \mathcal{C}$. The flying distance from location u to location v is denoted as $k_{u,v}$, where u and v are indexes of a set of locations $(\mathcal{C} \cup \mathcal{P})$. Let $\Omega = \{\omega_1, \omega_2, \dots, \omega_{\omega'}\}$ be a set of takeoff condition scenarios, where ω' represents the total number of scenarios in the set. Each ω is defined as $\omega = \{\mathbb{R}_1, \mathbb{R}_2, \dots, \mathbb{R}_{d'}\}$, where the subscript indicates the drone identification. Again, d' represents the total number of drones. $\mathbb{R}_d = 1$ when drone d cannot take off from the depot, and $\mathbb{R}_d = 0$ otherwise. Let $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_{\lambda'}\}$ be a set of breakdown condition scenarios, where λ' denotes the total number of the scenarios. Each λ is a parameter matrix of $\mathbb{B}_{i,d}$, where $\mathbb{B}_{i,d} = 1$ when drone d breaks down while serving customer i , and $\mathbb{B}_{i,d} = 0$ otherwise.

To minimise the total delivery cost, which is one of the objectives of the MODD framework, I consider four costs including (i) the initial cost of drones, i.e., $\mathfrak{C}^{(i)}$, (ii) the operational cost, i.e., $\mathfrak{C}^{(r)}$, (iii) the package transfer cost from an original depot to the new depot, i.e., $\mathfrak{C}^{(t)}$, and (iv) the outsourcing cost, i.e., $\mathfrak{C}^{(c)}$.

The summary of the notations in the MODD framework is presented in Table 4.1.

4.2 Multi-objective optimisation formulations

This section presents the optimisation problem formulations of the MODD framework. The detail of the three objective functions and method to solve the multi-objective optimisation problem are presented in Section 4.2.1. I define the decision variables and the constraints of the MODD framework in Section 4.2.2 and 4.2.3, respectively. At the end, Section 4.2.4 presents the solution technique.

TABLE 4.1: The notations in the MODD framework

Notations	Meaning
$\mathcal{C}$	Set of customers. i and j are indexes
$\mathcal{D}$	Set of drones
$\mathcal{F}$	Set of time windows
$\mathcal{P}$	Set of depots
Ω	Set of takeoff condition scenarios
Λ	Set of breakdown condition scenarios
a_i	The package weight of customer i
g_d	The capacity limit of drone d
e_d	The flying distance limit per trip of drone d
l_d	The flying distance limit per day of drone d
$\hat{h}_d$	The start flying time of drone d
$\bar{h}_d$	The end flying time of drone d
s_d	The average flying speed of drone d
$\hat{\mathbb{T}}_i^{(f)}$	The start of the time window that customer i prefers as the f^{th} order
$\bar{\mathbb{T}}_i^{(f)}$	The end of the time window that customer i prefers as the f^{th} order
t_i	The serving time of customer i
$o_{i,p}$	The indicator of packages belongs to which depot. $o_{i,p} = 1$ when the package of customer i belongs to depot p , and $o_{i,p} = 0$ otherwise
$k_{u,v}$	the flying distance from location u to location v
$\mathfrak{C}^{(i)}$	The initial cost of drones
$\mathfrak{C}^{(r)}$	The operational cost
$\mathfrak{C}^{(t)}$	The package transfer cost from an original depot to the new depot
$\mathfrak{C}^{(c)}$	The outsourcing cost
$P(\omega)$	The probabilities of the takeoff scenario ω
$P(\lambda)$	The probabilities of the breakdown scenario λ

4.2.1 Multiple objective functions

There are three objective functions in the MODD framework, i.e., (i) to minimise the total delivery cost, (ii) to minimise the percentage of unsuccessful delivered packages, and (iii) to maximise the reward of on-time delivery. The formulation of these objectives are presented in (4.1), (4.2), and (4.3), respectively. $\mathbb{O}^C$ represents the total delivery cost, $\mathbb{O}^U$ represents the percentage of unsuccessful delivered packages, and $\mathbb{O}^R$ represents the reward of on-time delivery.

$$\text{Minimise: } \mathbb{O}^C = \sum_{d \in \mathcal{D}} \mathfrak{C}_d^{(i)} W_d + \sum_{i \in \mathcal{C}, d \in \mathcal{D}, p \in \mathcal{S}} \left(\mathfrak{C}_{p,i}^{(r)} + \mathfrak{C}_{i,p}^{(r)} \right) Y_{i,d,p} + \sum_{p \in \mathcal{S}} \mathfrak{C}_p^{(t)} T_p + \sum_{i \in \mathcal{C}} \mathfrak{C}_i^{(c)} Z_i \quad (4.1)$$

$$\text{Minimise: } \mathbb{O}^U = \frac{100}{c'} \sum_{\substack{\omega \in \Omega \\ i \in \mathcal{C} \\ d \in \mathcal{D}}} \left(\mathbb{P}(\omega) X_{i,d}^{(b)}(\omega) + \sum_{\lambda \in \Lambda} \mathbb{P}(\omega, \lambda) X_{i,d}^{(a)}(\omega, \lambda) \right) \quad (4.2)$$

$$\text{Maximise: } \mathbb{O}^R = \sum_{i \in \mathcal{C}, f \in \mathcal{F}} C^{(f)} F_i^{(f)} \quad (4.3)$$

Note that $C^{(f)}$ represents the constant parameter, where $C^{(f_1)} > C^{(f_2)} > \dots > C^{(f_{f'})}$. Again, f_1 is the most preferred time window and $f_{f'}$ is the least preferred time window.

4.2.2 Decision variables

There are thirteen decision variables in the MODD framework. All the decision variables are binary, except $U_{i,d}$ which is an integer, and Q_i which is a positive variable. The definitions of the decision variables are as follows.

- W_d is the indicator for determining whether drone d is used or not, i.e., if $W_d = 1$, drone d will be used in the delivery, and $W_d = 0$ otherwise.
- $Y_{i,d,p}$ is the allocation variable. If $Y_{i,d,p} = 1$, customer i will be served by drone d , and the drone will depart from depot p , and $Y_{i,d,p} = 0$ otherwise.
- Z_i is the indicator for determining whether the package of customer i will be delivered by a carrier or not. If $Z_i = 1$, customer i will be served by the outsourcing carrier, and $Z_i = 0$ otherwise.
- T_p is the indicator for determining whether whether the shipper has to transfer packages from/to depot p or not. $T_p = 1$ means that at least one package is transferred from/to depot p , and $T_p = 0$ otherwise.
- $M_{i,p,q}$ is the indicator for determining whether the package of customer i is transferred from depot p to depot q or not. If $M_{i,p,q} = 1$, the package of customer i is transferred from depot p to depot q , and $M_{i,p,q} = 0$ otherwise.
- $B_{d,p}$ is an auxiliary variable for imposing the drone to have only one departing and returning depot.

- $U_{i,d}$ is a serving order of drone d , where $U_{i,d} < U_{j,d}$ means that drone d will serve customer i before customer j .
- $A_{i,j,d}$ is an auxiliary variable for ensuring that two customers cannot have the same serving order when they are served by the same drone.
- $X_{i,d}^{(b)}(\omega)$ is the takeoff condition variable in which $X_{i,d}^{(b)}(\omega) = 1$ means that drone d cannot takeoff to serve customer i under scenario ω , and $X_{i,d}^{(b)}(\omega) = 0$ otherwise.
- $X_{i,d}^{(a)}(\omega, \lambda)$ is the breakdown condition variable in which $X_{i,d}^{(a)}(\omega, \lambda) = 1$ means that drone d breaks down during serving customer i under takeoff scenario ω and breakdown scenario λ . Otherwise, $X_{i,d}^{(a)}(\omega, \lambda) = 0$.
- $F_i^{(f)}$ is the indicator for determining whether the f^{th} preferred time window is selected or not. $F_i^{(f)} = 1$ means that customer i will be served during the time window that the customer prefers as the f^{th} order, and $F_i^{(f)} = 0$ otherwise.
- $V_{i,j,d}$ is the indicator for determining whether drone d serves customer i , and then it serves customer j as the next customer or not. If $V_{i,j,d} = 1$, drone d will serve customer i before customer j , and $V_{i,j,d} = 0$ otherwise.
- Q_i is a serving time variable. If $Q_i \geq Q_j$, customer i will be served before customer j .

4.2.3 Constraints

There are four groups of the constraints in the MODD framework including (i) general constraints for the drone delivery with package transfer, (ii) breakdown constraints for the drone delivery, (iii) the serving order constraints for the drone delivery, and (iv) time window constraints with the reward counter. The general constraints for the drone delivery that I use in this chapter are similar to the constraints in (2), (3), (5), and (7) to (16) of our previous work in [122]. The constraints include the initial cost constraint, package allocation constraint, package transfer constraints, traveling time limit constraints, traveling distance limit

constraints, and capacity constraint. The formulations and the explanations of (ii), (iii), and (iv) are presented as follows:

$$\sum_{p \in \mathcal{S}} Y_{i,d,p} \mathbb{R}_d(\omega) - Z_i = X_{i,d}^{(b)}(\omega), \quad \forall i \in \mathcal{C}, d \in \mathcal{D}, \omega \in \Omega \quad (4.4)$$

$$\sum_{p \in \mathcal{S}} Y_{i,d,p} (1 - \mathbb{R}_d(\omega)) \mathbb{B}_{i,d}(\lambda) \leq X_{i,d}^{(a)}(\omega, \lambda), \quad \forall i \in \mathcal{C}, d \in \mathcal{D}, \omega \in \Omega, \lambda \in \Lambda \quad (4.5)$$

$$U_{i,d} - U_{j,d} \leq \Delta \left(1 - X_{j,d}^{(a)}(\omega, \lambda) + X_{i,d}^{(a)}(\omega, \lambda) \right), \quad i \neq j, \forall i, j \in \mathcal{C}, d \in \mathcal{D}, \omega \in \Omega, \lambda \in \Lambda \quad (4.6)$$

The constraints in (4.4) and (4.5) ensure that the percentage of unsuccessful delivered packages is calculated from the number of packages that cannot be delivered when the drone cannot take off and when the drone breaks down during serving, respectively. The constraint in (4.6) ensures that the rest of the packages are taken into account in the number of unsuccessful delivered packages after the drone breaks down. For example, $U_{j,d} > U_{i,d}$ and drone d breaks down while serving customer i , drone d will not be able to deliver the package of customer j .

$$\sum_{p \in \mathcal{S}} Y_{i,d,p} \leq U_{i,d}, \quad \forall i \in \mathcal{C}, d \in \mathcal{D} \quad (4.7)$$

$$0 \leq U_{j,d} \leq \sum_{i \in \mathcal{C}, p \in \mathcal{S}} Y_{i,d,p}, \quad \forall j \in \mathcal{C}, d \in \mathcal{D} \quad (4.8)$$

$$U_{i,d} - U_{j,d} \leq \Delta A_{i,j,d} - \sum_{p \in \mathcal{S}} Y_{i,d,p}, \quad i \neq j, \forall i, j \in \mathcal{C}, d \in \mathcal{D} \quad (4.9)$$

$$U_{i,d} - U_{j,d} \geq \sum_{p \in \mathcal{S}} Y_{i,d,p} - \Delta(1 - A_{i,j,d}), \quad i \neq j, \forall i, j \in \mathcal{C}, d \in \mathcal{D} \quad (4.10)$$

The constraints in (4.7) to (4.10) are the serving order constraints of drones. The constraints in (4.7) and (4.8) are the boundary constraints of serving order variables. The constraints in (4.9) and (4.10) ensure that the customers have the different serving order if they are served by the same drone. These constraints are

similar to those in chapter 3.

$$U_{j,d} - U_{i,d} \leq \Delta V_{i,j,d}, \quad i \neq j, \forall i, j \in \mathcal{C}, d \in \mathcal{D} \quad (4.11)$$

$$Q_i + \sum_{p \in \mathcal{S}} \left(\frac{k_{i,p} + k_{p,i}}{s_d} + t_i \right) Y_{i,d,p} - Q_j \leq \Delta(1 - V_{i,j,d}), \quad i \neq j, \forall i, j \in \mathcal{C} \quad (4.12)$$

$$\hat{h}_d \leq Q_i + Z_i \quad \forall i \in \mathcal{C}, d \in \mathcal{D} \quad (4.13)$$

$$Q_i \leq \bar{h}_d - \frac{k_{i,p} + k_{p,i}}{s_d} - t_i, \quad \forall i \in \mathcal{C}, d \in \mathcal{D} \quad (4.14)$$

$$\hat{\mathbb{T}}_i^{(f)} - Q_i + \sum_{p,d} \frac{k_{i,p} + k_{p,i}}{s_d} Y_{i,d,p} \leq \Delta(1 - F_i^{(f)}), \quad \forall i \in \mathcal{C}, f \in \mathcal{F} \quad (4.15)$$

$$Q_i + \sum_{p,d} \left(\frac{k_{i,p} + k_{p,i}}{s_d} + t_i \right) Y_{i,d,p} - \bar{\mathbb{T}}_i^{(f)} \leq \Delta(1 - F_i^{(f)}), \quad \forall i \in \mathcal{C}, f \in \mathcal{F} \quad (4.16)$$

$$\sum_{f \in \mathcal{F}} F_i^{(f)} \leq 1, \quad \forall i \in \mathcal{C} \quad (4.17)$$

The constraints in (4.11) to (4.14) ensure that (i) serving time $Q_i < Q_j$ with the serving order $U_i < U_j$, and (ii) the time between Q_i and Q_j must be longer than the time of flying from customer i to the depot plus flying from the depot to customer j . The constraints in (4.15) and (4.16) ensure that serving time Q_i is between the period of the selecting time window. For example, if the most preferred time window ($f = 1$) of customer i is selected, i.e., $F_i^{(1)} = 1$, then $\hat{\mathbb{T}}_i^{(1)} \leq Q_i \leq \bar{\mathbb{T}}_i^{(1)}$. Again, $\hat{\mathbb{T}}_i^{(1)}$ and $\bar{\mathbb{T}}_i^{(1)}$ are the start and the end of the time window, respectively. The constraint in (4.17) ensures that only one time window can be selected.

4.2.4 Solution Technique: ϵ -constraint method

To solve the problem, I convert the proposed multi-objective optimisation to a linear programming by the ϵ -constraint method [30]. I can use one of the three objectives in (4.1) to (4.3) as the objective function and use the others as constraints. An example of the conversion are as follows:

Minimise: $\mathbb{O}^C$

subject to (4.4) to (4.17), (4.18), and (4.19), .

$$\mathbb{O}^U \leq \epsilon_1 \quad (4.18)$$

$$\mathbb{O}^R \geq \epsilon_2. \quad (4.19)$$

The effective solutions of the MODD framework can be achieved by vary the parameters ϵ_1 and ϵ_2 . Again, the objective function can be to minimise $\mathbb{O}^C$, to minimise $\mathbb{O}^U$, or to maximise $\mathbb{O}^R$. Once the objective is selected, the other objectives are taken as the constraints with parameters ϵ_k . The steps of solving the multi-objective optimisation by ε -constraints are listed below.

Step 1 Initialise the range of ϵ_k parameters [42].

Step 2 Solve the linear programming problem by using parameter ϵ with the smallest value.

Step 3 Increase the value of one of the parameters ϵ_k and repeat Step 2. Once the value of parameter ϵ_k is not in the range, the algorithm is terminated.

Step 4 Obtain the set of solutions.

Note that the best solution is not selected for the shipper because the different shipper may have different criterion for the multi-objective optimisation. This chapter presents the feasible solutions for the shipper to select.

Next, I evaluate the MODD framework and present the experimental results.

4.3 Performance evaluation

In this section, I evaluate the MODD framework with the real customer data from a Singapore logistic company. Forty customers are considered in all the experiments. Each customer has a location, the start of time window, and the end of time window. I assume that all the packages of customers are $a_i \leq 5$ kg. Therefore, the cost of outsourcing package to a carrier is set as $\mathfrak{C}^{(c)} = S\$16$ based on the Speedpost service of SingPost company. The shipper has two depots, i.e., $\mathcal{P} = \{p_1, p_2\}$, and two drones, i.e., $\mathcal{D} = \{d_1, d_2\}$. The drones are of the same type, where $\mathfrak{C}_d^{(i)} = S\100 , $l_d = 150$ km, $e_d = 10$ km, $g_d = 5$ kg, $h_d = 8$ hrs, and $s_d = 30$ km/hr. The package transferring cost of both depots is set as $\mathfrak{C}^{(t)} = S\$30$. I assume that routing of drones is similar to ground-based vehicle because Singapore has many high buildings, e.g., residential housing. The operational cost can be calculated by $\mathfrak{C}^{(r)} = distance \times 1.05 \times 0.1$, where the distance in kilometres between one location

FIGURE 4.1: Map

and the other location is extracted from Google Map. Two takeoff and breakdown scenarios are considered in all the experiments, i.e., $\Lambda = \{\lambda_1, \lambda_2\}$ and $\Omega = \{\omega_1, \omega_2\}$. Let λ_1 be the scenario that all the drones can take off and λ_2 be the scenario that all the drones cannot take off, e.g. due to raining. Let ω_1 be the scenario that breakdown does not occur and ω_2 be the scenario that breakdown occurs every time when the drones serve customers. The probabilities of the scenarios are set as $P(\lambda_1) = P(\omega_1) = 0.9$ and $P(\lambda_2) = P(\omega_2) = 0.1$. Note that the probabilities can be calculated based on the history record. The GAMS script is implemented for the optimisation problem and is solved by the solver CPLEX [22].

4.3.1 Pareto frontier

I experiment the framework with two objectives, which are to minimise the total cost, i.e., $\mathbb{O}^C$, and to minimise the percentage of unsuccessful delivered packages, i.e., $\mathbb{O}^U$. The Pareto frontier of these two objectives is presented in Figure 4.2. To guarantee that the shipper will not fail to deliver any packages, the shipper can outsource all packages to the carrier, which is more expensive than delivering by using the shipper's own drones. The total cost is stable when the percentage of unsuccessful delivered packages is more than 13.325%. For this experiment setting, the shipper needs to pay at least S\$415.16 to deliver all packages.

FIGURE 4.2: The Pareto frontier

4.3.2 Reward of on-time delivery

To demonstrate the impact of the reward of on-time delivery, I present two test cases, i.e., (i) minimising the total delivery cost in Figure 4.3a and (ii) minimising the percentage of unsuccessful delivered packages in Figure 4.3b. Note that if I use (i) or (ii) as an objective function, then (ii) and (i) will become a constraint, respectively.

From Figure 4.3a, fifteen is reached as the highest number of matching time windows by using two drones to serve all the customers in the coverage area of the drones. When the reward of on-time delivery is larger than or equal to 17, the shipper has not enough drones to satisfy the time window requirement of all the customers. The total cost is higher when I force the percentage of unsuccessful delivered packages to be lower than a certain value as the shipper needs to outsource some packages to reduce the breakdown event.

From Figure 4.3b, to minimise the percentage of unsuccessful delivered packages, the shipper can outsource as many packages as possible to the carrier. Consequently, these packages will be delivered without experiencing the drone breakdown. The shipper can use one drone to serve the customers when the total cost needs to be lower than S\$450. When the total cost constraint is less than or equal to S\$450, the percentage of unsuccessful delivered packages is higher than that when

(A) minimising the total cost

(B) minimising the percentage of unsuccessful delivered packages

FIGURE 4.3: Impact of number of matching required time windows while minimising (a) the total cost and (b) the percentage of unsuccessful delivered packages.

FIGURE 4.4: The impact of breakdown probability on the percentage of unsuccessful delivered packages.

the total cost constraint is less than or equal to $S\$550$. The reason is that the number of outsourced packages is fewer. If the shipper does not consider the total cost constraint, the percentage of unsuccessful delivered packages varies linearly with respect to the reward of on-time delivery.

4.3.3 Impact of probabilities (λ and ω)

The takeoff conditional probability is set equal to the breakdown condition probability, i.e., $P(\lambda_1) = P(\omega_1)$ and vary them. For ease of the presentation, I do not consider the reward of on-time delivery in this experiment.

When the total cost is less than or equal to $S\$450$, 23 customers are served by a drone, and the rest are served by the carrier. Similarly, 20 customers, 27 customers, and 30 customers are served by the drone when the total cost is less than or equal to $S\$500$, $S\$550$, and $S\$600$, respectively. From Figure 4.4, I can conclude that the percentage of unsuccessful delivered packages varies proportionally to the breakdown probability, except when the total cost is larger than or equal to $S\$640$. When the total cost is larger than or equal to $S\$640$, the shipper will not experience the drone breakdown because the shipper can outsource all packages to the carrier, i.e., $(40 \times S\$16)$, which will not incur the percentage of unsuccessful delivered packages.

4.4 Conclusion

The multi-objective optimisation have been proposed for drone delivery (MODD) framework to help the shipper schedule and plan its delivery by providing a set of potential solutions. I have formulated three different objectives in the framework, i.e., to minimise the total delivery cost, to minimise the percentage of unsuccessful delivered packages, and to maximise the reward of on-time delivery ([Objective 4](#)). The trade-off between using drones and outsourcing packages to a carrier has been considered in the framework as drones may not be able to reach some customers, e.g., due to flying distance limit, and thus [Objective 1](#) is achieved. Furthermore, I have formulated the MODD framework as a three-objective and three-stage stochastic programming. The takeoff and breakdown conditions from [Chapter 3](#) are taken into account during the second and the third stages of the optimisation, respectively. As a result, [Objective 2](#) is automatically revisited. I have evaluated the MODD framework with the real data from a Singapore company, and I presented the analysis of the Pareto frontiers of the MODD framework.

Chapter 5

Drone and Outsource Delivery Planning for Multiple Shippers ¹

Extending from Chapter 4, drone delivery is expected to be combined with other delivery modes such as outsourcing package to a third-party carrier. This combined delivery becomes an appropriate option for shippers to meet customers' needs while maintaining low cost and high reliability.

Alternatively, logistics business partnership becomes a viable choice of reducing cost and improving delivery efficiency. Multiple shippers can cooperate and form a pool of delivery resources including depots, manpower, and drones [48]. For example, a cooperative shipper can use vehicles to serve its customers and when the vehicles are not used, they can be utilised by the other cooperative shippers. This opportunistic logistics resource sharing can clearly improve the resource utilisation and hence profitability.

With the cooperation among shippers with drones, the shippers need to answer the following questions: (i) should the shippers cooperate with other shippers? (ii) which shippers should cooperate with? (iii) how can the shippers with cooperation absorb the delivery cost in a fair manner? and (iv) how many drones are required and how to assign packages to the drones or to use an outsource carrier. To address these questions, I therefore propose the Bayesian Shipper Cooperation in Stochastic Drone Delivery (BCoSDD) framework for helping shippers to strategise their delivery planning and resource sharing [122]. The BCoSDD framework is

¹The work in this chapter has been presented in [125]

composed of three components, i.e., package assignment, shipper cooperation, and cost management.

According to the objectives in Section 1.2, [Objective 2](#), [Objective 5](#) and [Objective 6](#) are accomplished in this chapter. This chapter is the extension of the GADOP and MODD framework. The uncertainties in drone delivery, i.e., breakdown condition, is extended from the three-stage stochastic optimisation in the GADOP and MODD frameworks into a multi-stage stochastic optimisation in this chapter. Thus, [Objective 5](#) is revisited. Again, the contributions of this chapter are summarised as follows.

Objective 2: In the proposed BCoSDD framework, I also introduce the package assignment based on multi-stage stochastic programming optimisation. The package assignment takes the uncertainty of the drone breakdown into account to assign packages to be delivered by drones with the lowest cost while meeting all customers' demand. While the original problem of the package assignment is non-linear, the problem is reformulated into a linear one that can be solved more efficiently.

Objective 5: In the proposed BCoSDD framework, I propose a shipper cooperation management. The static Bayesian coalition formation game is proposed for the case that the belief about the shippers is known a priori or assumed that the shippers always deliver packages as it is assigned. I analyse the stability of the shipper cooperation analytically and also present an algorithm to reach the stable solution. The algorithm to decide which the shippers should cooperate is summarised in [Algorithm 2](#).

Moreover, among the shippers cooperating with each other, the incurred cost will be distributed. I therefore propose cost management scheme to allow the cooperative shippers to share the cost of package delivery. The Shapley value technique is adopted to reach a fair cost sharing.

Objective 6: Alternatively, I propose the dynamic Bayesian coalition formation game for the case that the shippers have to observe and learn behaviours of the other shippers without complete belief. In reality, some shippers may misbehave in the cooperation, e.g., a shipper may not deliver a package as it is assigned. Consequently, the algorithm is thus designed to learn the shippers'

behaviour and allow honest shippers to form the cooperation effectively. The algorithm of the belief update machine is presented in Algorithm 3.

In summary, the proposed BCoSDD framework is useful for the shippers not only for their logistics resource and delivery planning, but also for reaching an optimal business partnership through cooperation to share the resources efficiently. The ultimate goals are to achieve the minimum cost and stable cooperation strategy by the shippers.

This chapter is organised as follows. Section 5.1 describes the system model, which consists of three components, i.e., package assignment, shipper cooperation, and cost management. The package assignment and the shipper cooperation are discussed in detail in Section 5.2 and Section 5.4, respectively. After that, the dynamic algorithm for shipper cooperation formation is presented in Section 5.5. Section 5.6 explains the experiment setting and gives performance evaluation results of the proposed framework.

5.1 System model and assumptions

This section presents the proposed BCoSDD framework in detail. The framework considers multiple shippers, and each shipper is rational and self-interested. All the shippers have the same objective that is to minimise their delivery cost while the packet delivery must be done in a specific time slot, e.g. all packages are delivered by the end of the day. The shippers can use drones to deliver a package to a customer or outsource a package delivery to a third-party carrier. The drone flies from a depot to serve the customer and returns to the depot. If a shipper outsources the package delivery to a carrier, the shipper pays a fixed cost, i.e., a fee, to the carrier. When a shipper or a coalition of shippers deliver multiple packages, the delivery cost is assumed to be cheaper than that of outsourcing package delivery. Nonetheless, outsourcing package delivery to a carrier is still needed to meet customers' demand in some cases, for example, when drones cannot serve customers which are not in the flying coverage area.

Furthermore, the shippers can cooperate and create a resource pool to minimise their delivery cost. Once the resource pool is created, a shipper can let other

cooperative shippers in the same pool deliver a package. In this case, the package can be transferred from one depot to another depot of different cooperative shippers to facilitate the use of drone for delivery, i.e., drone sharing. With the resource pool, the incurred cost will be distributed by the cooperative shippers forming the same pool in a fair manner.

The BCoSDD framework is composed of three components as shown in Figure 5.1. I first address the package assignment component for an individual shipper or a coalition of cooperative shippers, also called a coalition of shippers, by an optimisation technique. Next, I present the cost management component which is done after the solution of the package assignment problem is obtained. After that, the shipper cooperation component is used to decide how the shippers should cooperate and form a coalition and a resource pool. The interactions among these components are as follows.

- First, given a set of coalitions of cooperative shippers, i.e., a coalition structure, the package assignment is performed to assign packages to drones or to outsource packages to a carrier.
- Second, the optimal solution of the package assignment is then used by the cost management to divide the package delivery cost among the cooperative shippers in the same group.
- Third, given the cost divided among the cooperative shippers, the shipper cooperation is performed, i.e., through the merge-and-split algorithm, to achieve the stable coalition structure.

FIGURE 5.1: The shipper cooperation in static BCoSDD framework.

5.2 Package assignment

In this section, the package assignment is discussed in the detail. Firstly, the background of package assignment is presented. Secondly, how the package assignment process works is introduced. Then, the formulations of the package assignment optimisation are presented.

5.2.1 Package assignment background

In this chapter, package assignment refers to the optimisation process of allocating customers' packages to cooperative shippers, and for each shipper, which delivery mode, i.e., drone or outsource, is used. The cooperative shippers contribute a set of drones and a set of customers to the coalition. Again, coalition refers to the resource pool that the shippers establish when the shippers cooperate. Each drone that the shippers share to the coalition may have different operation cost. Each shipper has a depot, which is located in different area. The package assignment allows the shippers to transfer customer's package from one depot to the other depot before delivery is performed, and the transferring fee is considered in the total delivery cost.

The solution of the package assignment is collective in that the assignment is done for an entire coalition to which the total delivery cost of the coalition is minimised. Each customer package is allocated to either the shipper drone or an outsource carrier. The solution also includes whether customers' packages need to be transferred to the new depot or not. The routing of drone is also determined. Moreover, due to uncertainty of the drone delivery, the solution of package assignment will be multi-stage. If any breakdown occurs to a certain drone, all the packages which are assigned to the drone will be delivered by an outsourcing carrier with a high penalty cost. The total delivery cost will be shared among the cooperative shippers in the same coalition in a fair manner based on their contribution to the package assigned to be delivered. The cost sharing is processed in the other component, i.e., cost management, which is presented in Section 5.3.

5.2.2 Package delivery operation and scenario tree

Let $\mathcal{P} = \{p_1, p_2, \dots, p_{|\mathcal{P}|}\}$ be a set of all shippers in the system, where $|\mathcal{P}|$ denotes the total number of shippers. Each shipper p has one depot, a set of customers denoted by C_p , and a set of drones denoted by D_p . Without loss of generality, each customer has one package to be delivered. Note that if a customer has two or more packages to be delivered, I can consider that there are multiple customers located at the same location.

The package assignment is executed for one given coalition of cooperative shippers. Let $\mathcal{S}$ be a set of shippers that cooperate with each other, creating a resource pool and forming a coalition $\mathcal{S}$, i.e. $\mathcal{S} \subset \mathcal{P}$. The shippers in the same coalition allow their drones to deliver packages of each other. In this case, a shipper can transfer its packages from its depot to the depots of other shippers. Accordingly, the packages of the customers of the shippers in the same coalition will be delivered by any of the drones. For example, for $\mathcal{S} = \{p_1, p_2, p_3\}$ i.e., shippers p_1 , p_2 , and p_3 cooperate and form a coalition, the set of customers for this coalition is $\mathcal{C} = C_1 \cup C_2 \cup C_3$ and the set of drones in this coalition is $\mathcal{D} = D_1 \cup D_2 \cup D_3$. In other words, $\mathcal{C} = \{c_1, c_2, \dots, c_{|\mathcal{C}|}\}$ and $\mathcal{D} = \{d_1, d_2, \dots, d_{|\mathcal{D}|}\}$ denote the sets of customers and drones in the pool, where $|\mathcal{C}|$ and $|\mathcal{D}|$ represent the total number of customers and the total number of drones, respectively. As the drones are shared, their utilisation and efficiency can be increased. Note that if the shipper does not cooperate with any other shippers, the coalition contains only one shipper in which the same package assignment still applies.

I assume that a drone can depart and return to the same depot only and cannot switch the depot. This is, for example, due to the flying program embedded in the drone. However, although the reliability of drones has been improved significantly, they are still subject to breakdown and failure, e.g., from bad weather conditions. A drone breakdown happens randomly. Thus, when a drone delivers a package to a customer, it can be successful or fail. If there is no breakdown, the drone will continue to deliver to the next customer until all the assigned packages are delivered. However, if there is a breakdown, the drone cannot be used to deliver the rest of the assigned packages. In this case, the shippers have to pay penalty such as compensation to the customers of the rest of the assigned packages. Note that shippers can outsource the package delivery to a third-party carrier. However, the cost of outsourcing is typically higher than that of using their drones. Nonetheless,

FIGURE 5.2: An example of a scenario tree of a drone.

outsourcing can still be used, for example, if the chance of drone breakdown is high because of looming bad weather conditions.

5.2.3 Package assignment formulations

Extending from Chapter 3 and Chapter 4, the package assignment problem is formulated as a multi-stage instead of the three-stage stochastic programming optimisation. Figure 5.2 shows a scenario tree example of the stochastic model for one drone. In the first stage, the following activities happen, i.e., the drone is reserved for the delivery, selected packages are transferred to different depots, and the shippers outsource the package delivery to the carrier. From the second stage, each of the stages is associated with the delivery of one package of a customer sequentially in which the total number of stages is $|c|^{\text{th}}$. At the transition from the current stage to the next stage, there are two possible events, i.e., the delivery is successful or fails, which is represented by a branch. If the delivery is successful, the next stage ensues. However, if the delivery fails, the penalty is paid and the transition terminates as the drone cannot be used to deliver the rest of the assigned packages. Here, $\mathbb{P}_d$ denotes a breakdown probability of drone d , which can be calculated based on historical data.

The objective of the package assignment is to minimise the total delivery cost of a coalition of cooperative shippers, i.e., $\mathbb{O}^{ST}$. The multi-stage stochastic programming formulation is expressed as follows:

Minimise:

$$\mathbb{O}^{\text{ST}}(\mathcal{S}) = \mathbb{O}^{\text{DE}} + \sum_{d \in \mathcal{D}} \sum_{j=1}^{N_d} ((1 - \mathbb{P}_d)^{j-1} \mathbb{P}_d C^{(p)}(N_d - j + 1)), \quad (5.1)$$

where

$$\mathbb{O}^{\text{DE}} = \sum_{d \in \mathcal{D}} C_d^{(i)} W_d + \sum_{i \in \mathcal{C}, d \in \mathcal{D}, p \in \mathcal{S}} (C_{p,i}^{(r)} + C_{i,p}^{(r)}) Y_{i,d,p} + \sum_{p \in \mathcal{S}} C_p^{(t)} T_p + \sum_{i \in \mathcal{C}} C_i^{(c)} Z_i, \quad (5.2)$$

subject to (5.3) - (5.16). Recall that the optimisation formulation of the package assignment is for a given coalition $\mathcal{S}$. Nonetheless, I omit $\mathcal{S}$ in the expressions and notations for the rest of this section to simply the presentation.

The delivery cost as expressed in (5.1) is composed of

- the initial cost of drone d , i.e., $C_d^{(i)}$,
- the operational cost of drone d to travel from the depot of shipper p to customer i , i.e., $C_{p,i}^{(r)}$,
- the cost of shipper p transferring packages among depots of the resource pool, i.e., $C_p^{(t)}$,
- the outsourcing cost per package, i.e., $C_i^{(c)}$ paid to a carrier, and
- the penalty cost when the drones fail to deliver a package, i.e., $C^{(p)}$.

There are seven decision variables in the formulation, including W_d , $Y_{i,d,p}$, Z_i , T_p , $M_{i,p,q}$, $B_{d,p}$, and N_d . The definitions of these decision variables are as follows.

- W_d is the indicator whether drone d is used or not, i.e., if $W_d = 1$, drone d will be used in the delivery, and $W_d = 0$ otherwise.
- $Y_{i,d,p}$ is the allocation variable. If $Y_{i,d,p} = 1$, customer i will be served by drone d , and the drone will depart from depot p , and $Y_{i,d,p} = 0$ otherwise.
- Z_i is the indicator whether customer i will be served by a drone or not. If $Z_i = 1$, customer i will not be served by any drone, and $Z_i = 0$ otherwise.

- T_p is the indicator whether shipper p has to transfer packages from/to other shippers or not. If $T_p = 1$, shipper p will transfer packages to another cooperative shipper, and $T_p = 0$ otherwise.
- $M_{i,p,q}$ is the indicator whether shipper p transfers the package of customer i to shipper q or not. If $M_{i,p,q} = 1$, shipper p transfers the package of customer i to shipper q , and $M_{i,p,q} = 0$ otherwise.
- $B_{d,p}$ is an auxiliary variable for imposing a drone to have only one departure and returning depot.
- N_d is the total number of customers that will be served by drone d .

The package assignment is subject to the constraints in (5.3) - (5.16).

$$\sum_{i \in \mathcal{C}, p \in \mathcal{S}} Y_{i,d,p} \leq \Delta W_d, \quad \forall d \in \mathcal{D} \quad (5.3)$$

$$\sum_{i \in \mathcal{C}, q \in \mathcal{S}} M_{i,p,q} \leq \Delta T_p, \quad \forall p \in \mathcal{S} \quad (5.4)$$

$$\sum_{i \in \mathcal{C}, q \in \mathcal{S}} M_{i,q,p} \leq \Delta T_p, \quad \forall p \in \mathcal{S} \quad (5.5)$$

$$a_i \sum_{p \in \mathcal{S}} Y_{i,d,p} \leq f_d, \quad \forall i \in \mathcal{C}, d \in \mathcal{D} \quad (5.6)$$

$$\sum_{d \in \mathcal{D}, p \in \mathcal{S}} Y_{i,d,p} + Z_i = 1, \quad \forall i \in \mathcal{C} \quad (5.7)$$

$$o_{i,p} - \sum_{q \in \mathcal{S}} M_{i,p,q} \leq \Delta \sum_{d \in \mathcal{D}} (Y_{i,d,p} + Z_i), \quad \forall i \in \mathcal{C}, p \in \mathcal{S} \quad (5.8)$$

The constraint in (5.3) ensures that the initial cost of a drone is paid when any packages are assigned to the drone. Note that Δ denotes a large number. The constraints in (5.4) and (5.5) ensure that the package transferring cost is paid when shippers transfer packages to the other shippers. The constraint in (5.6) ensures that the total weight of a package to be delivered by a drone does not exceed the capacity limit of the drone, where drone d can carry a package with the weight up to f_d kilograms. The constraints in (5.7) and (5.8) ensure that all the customer packages are assigned to a drone, or otherwise the outsourcing cost needs to be paid. I use variable $o_{i,p} = 1$ to define that customer i belongs to shipper

p , and $o_{i,p} = 0$ otherwise. I have the condition $\sum_{p \in \mathcal{P}} o_{i,p} = 1$ for all $i \in \mathcal{C}$. If $o_{i,p} = 1$, a drone that will deliver the package of customer i must depart from the depot of shipper p . Otherwise, the package must either be transferred to a new depot or outsourced to a carrier. The constraints in (5.9) to (5.11) ensure that the drone does not travel exceeding the coverage area, traveling distance limit, and traveling time limit, respectively. Here, t_i is the serving time, i.e., the time that the drone spends to search and drop a package at the destination. The drone has flying distance limits, which are up to e_d kilometers per trip and l_d kilometers per day. The average flying speed of drone d is denoted as s_d . Let $k_{i,j}$ denote the flying distance from location i to location j .

$$\sum_{p \in \mathcal{S}} Y_{i,d,p} (k_{p,i} + k_{i,p}) \leq e_d, \quad \forall i \in \mathcal{C}, d \in \mathcal{D} \quad (5.9)$$

$$\sum_{i \in \mathcal{C}, p \in \mathcal{S}} Y_{i,d,p} (k_{p,i} + k_{i,p}) \leq l_d, \quad \forall d \in \mathcal{D} \quad (5.10)$$

$$\sum_{i \in \mathcal{C}, p \in \mathcal{S}} \left(\frac{k_{i,p} + k_{i,p}}{s_d} + t_i \right) Y_{i,d,p} \leq h_d, \quad \forall d \in \mathcal{D} \quad (5.11)$$

The constraints in (5.12) to (5.14) are the package transferring constraints. The constraint in (5.12) ensures that if a package is transferred to a new depot, the drone must depart from the new depot. The constraint in (5.13) ensures that a package can only be transferred once. The constraint in (5.14) ensures that no package is transferred to the original depot. The constraints in (5.15) and (5.16) ensure that a drone can only depart from and arrive to only one depot.

$$M_{i,p,q} \leq \sum_{d \in \mathcal{D}} Y_{i,d,q}, \quad \forall i \in \mathcal{C}, \forall p \in \mathcal{S}, \forall q \in \mathcal{S} \quad (5.12)$$

$$\sum_{p \in \mathcal{S}, q \in \mathcal{S}} M_{i,p,q} = 1, \quad \forall i \in \mathcal{C} \quad (5.13)$$

$$M_{i,p,p} = 0, \quad \forall i \in \mathcal{C}, \forall p \in \mathcal{S} \quad (5.14)$$

$$\sum_{i \in \mathcal{C}} Y_{i,d,p} \leq \Delta B_{d,p}, \quad \forall d \in \mathcal{D}, \forall p \in \mathcal{S} \quad (5.15)$$

$$\sum_{p \in \mathcal{S}} B_{d,p} = 1, \quad \forall d \in \mathcal{D} \quad (5.16)$$

$$\sum_{i \in \mathcal{C}, p \in \mathcal{S}} Y_{i,d,p} \leq N_d, \quad \forall d \in \mathcal{D} \quad (5.17)$$

$$0 \leq N_d \leq |\mathcal{C}|, \quad \forall d \in \mathcal{D} \quad (5.18)$$

The constraints in (5.17) and (5.18) limit the number of serving customers. The constraint in (5.17) calculates the total number of customers that are served by drone d , i.e., N_d . The constraint in (5.18) ensures that N_d is a positive integer and must be less than the total number of customers.

The solution of the package assignment can be obtained by solving the above multi-stage stochastic programming optimisation. However, it is a quadratic programming problem as the objective function is in a quadratic form. Since quadratic programming problems can be more complex to solve than linear programming problems, next I transform the quadratic programming problem to the linear programming problem.

5.2.4 Simplification

The quadratic programming problem in (5.1) is reformulated to the linear programming problem by introducing three auxiliary variables and three auxiliary constraints. The linear objective function is presented in (5.24).

Minimise:

$$\mathbb{O}^{\text{ST}} = \mathbb{O}^{\text{DE}} + \sum_{i \in \mathcal{C}, d \in \mathcal{D}} A_{i,d}, \quad (5.19)$$

subject to (5.3) - (5.18) and (5.25) - (5.28). $\mathbb{O}^{DE}$ can be calculated from (5.2).

The definitions of the three auxiliary variables are as follows.

- $A_{i,d}$ is an auxiliary positive variable. The value of $A_{i,d}$ will be a part of breakdown penalty, which is incurred by drone d .
- $X_{i,d}$ is an auxiliary binary variable. $X_{i,d} = 1$ when the order of i is less than or equal to the total number of drone d to be used in the delivery. Otherwise, $X_{i,d} = 0$.
- $V_{i,d}$ is an auxiliary positive variable. The value of $V_{i,d}$ will be a part of breakdown penalty, which is incurred by drone d .

$$\sum_{i \in \mathcal{C}} X_{i,d} = N_d, \quad \forall d \in \mathcal{D} \quad (5.20)$$

$$X_{i,d} \geq X_{j,d}, \quad \forall i, j \in \mathcal{C}, i < j, \forall d \in \mathcal{D} \quad (5.21)$$

$$V_{i,d} + \Delta X_{i,d} \leq \Delta + A_{i,d}, \quad \forall d \in \mathcal{D}, \forall i \in \mathcal{C} \quad (5.22)$$

The constraints in (5.25) and (5.26) ensure that the customer serving order does not exceed the total number of the customers. For example, if drone d serves five customers, i.e., $N_d = 5$, $X_{i,d} = 1$ for $i = 1, \dots, 5$, and $X_{i,d} = 0$ for $i > 5$. Note that $X_{i,d}$ is not related to a specific customer, and the set $\mathcal{C}$ and the index i are used for simplifying the parameter declaration. The constraint in (5.27) ensures that $A_{i,d} = V_{i,d}$ when i is the first N_d customers, i.e., $X_{i,d} = 1$ for $i = 1, \dots, N_d$. The constraint in (5.28) calculates the expected value of the penalty incurred by the drone breakdown.

$$V_{i,d} = \left(\prod_{j \in \mathcal{C}, j < i} (1 - \mathbb{P}) \right) \mathbb{P}^{C(p)} \left(N_d - \left(\sum_{j \in \mathcal{C}, j < i} 1 \right) + 1 \right), \forall d \in \mathcal{D}, i \in \mathcal{C} \quad (5.23)$$

The above optimisation can be solved by using a linear programming solver. W_d^* , $Y_{i,d,p}^*$, Z_i^* , T_p^* , $M_{i,p,q}^*$, $B_{d,p}^*$, N_d^* , $A_{i,d}^*$, $X_{i,d}^*$, and $V_{i,d}^*$ represent the optimal solutions

of W_d , $Y_{i,d,p}$, Z_i , T_p , $M_{i,p,q}$, $B_{d,p}$, N_d , $A_{i,d}$, $X_{i,d}$, and $V_{i,d}$, respectively. The optimal total delivery cost is denoted by $\mathbb{O}^{\text{ST}*}$ which is used in the cost management.

Again, the BCoSDD framework is composed of three components as discussed in Section 5.1 and summarised in Figure 5.1. The first component, which is package assignment, has been presented in this section. The next section presents the second component, i.e., cost management. The third component, i.e., shipper cooperation, is presented later.

5.2.4.1 The proof of the transformation

We reformulate the quadratic programming problem in Equation (5.1) to the linear programming problem by introducing one order set, three auxiliary variables, and four auxiliary constraints. Let $\mathcal{K} = \{1, 2, 3, \dots, |\mathcal{C}|\}$ be the set of possible serving orders of the customers, where the order cannot be larger than the total number of customers, i.e., $|\mathcal{C}|$. The linear objective function is presented in (5.24).

Minimize:

$$\mathbb{O}^{\text{ST}} = \mathbb{O}^{\text{DE}} + \sum_{k \in \mathcal{K}, d \in \mathcal{D}} A_{k,d}, \quad (5.24)$$

subject to and (5.25) - (5.28). $\mathbb{O}^{\text{DE}}$ can be calculated from (5.2). Note that more detail for the quadratic programming problem, i.e., the objective function presents in (5.2) and the constraints in (5.3) - (5.18). The constraints (5.3) - (5.18) remain the same after reformulation.

The definitions of the three introducing auxiliary variables are as follows.

- $A_{k,d}$ is an auxiliary positive variable. $A_{k,d} = 0$ when order k is larger than the total number of customers that will be served by drone d . Otherwise, $A_{k,d} = V_{k,d}$.
- $X_{k,d}$ is an auxiliary binary variable. $X_{k,d} = 1$ when the order k is less than or equal to the total number of customers that will be served by drone d . Otherwise, $X_{k,d} = 0$.
- $V_{k,d}$ is an auxiliary positive variable. The value of $V_{k,d}$ is the breakdown penalty, which is incurred by using drone d to serve customer order k .

$$\sum_{k \in \mathcal{K}} X_{k,d} = N_d, \quad \forall d \in \mathcal{D} \quad (5.25)$$

$$X_{k,d} \geq X_{k',d}, \quad \forall k, k' \in \mathcal{K}, k < k', \forall d \in \mathcal{D} \quad (5.26)$$

$$V_{k,d} + \Delta X_{k,d} \leq \Delta + A_{k,d}, \quad \forall d \in \mathcal{D}, \forall k \in \mathcal{K} \quad (5.27)$$

$$V_{k,d} = \left(\prod_{j \in \mathcal{K}, j < k} (1 - \mathbb{P}_d) \right) \mathbb{P}_d C^{(p)} \left(N_d - \left(\sum_{j \in \mathcal{K}, j \leq k} 1 \right) + 1 \right), \quad \forall d \in \mathcal{D}, k \in \mathcal{K} \quad (5.28)$$

The constraints in (5.25) and (5.26) ensure that the customer serving order does not exceed the total number of the customers. For example, if drone d serves five customers, i.e., $N_d = 5$, $X_{k,d} = 1$ for $k = 1, \dots, 5$, and $X_{k,d} = 0$ for $k > 5$. The constraint in (5.27) ensures that $A_{k,d} = V_{k,d}$ when k is the first N_d customers, i.e., $X_{k,d} = 1$ for $k = 1, \dots, N_d$. The constraint in (5.28) calculates the expected value of the penalty incurred by the drone breakdown.

The above optimization can be solved by using a linear programming solver. W_d^* , $Y_{i,d,p}^*$, Z_i^* , T_p^* , $M_{i,p,q}^*$, $B_{d,p}^*$, N_d^* , $A_{i,d}^*$, $X_{i,d}^*$, and $V_{i,d}^*$ represent the optimal solutions of W_d , $Y_{i,d,p}$, Z_i , T_p , $M_{i,p,q}$, $B_{d,p}$, N_d , $A_{i,d}$, $X_{i,d}$, and $V_{i,d}$, respectively. The optimal total delivery cost is denoted by $\mathbb{O}^{\text{ST}*}$ which is used in the cost management.

Since $\mathbb{O}^{DE}$ and the constraints in (5.3) - (5.18) are linear, the transformation only focuses on the penalty term, which are presented in (5.29).

$$\sum_{d \in \mathcal{D}} \sum_{k=1}^{N_d} ((1 - \mathbb{P}_d)^{k-1} \mathbb{P}_d C^{(p)}(N_d - k + 1)) \quad (5.29)$$

To calculate the value of the penalty term in (5.29), we need to know the value of decision variable N_d . Again, N_d is the total number of customers that will be served by drone d , which means that the value of N_d must be less than or equal to the total number of customers, i.e., $N_d \leq |\mathcal{C}|$. Therefore, the penalty term can be pre-calculated for all possible values of N_d . Note that the penalty term consists of

two summations. When we pre-calculate the penalty term, we consider only the value for each drone and each order (without the summations).

We have introduced $V_{k,d}$ as an auxiliary positive variable to store the pre-calculated penalty for all drones and all orders. The calculation of the pre-calculated penalty is presented as one of the auxiliary constraints, which is in (5.28). Again, $\mathcal{K} = \{1, 2, \dots, |\mathcal{C}|\}$ is the set of customer orders.

$$V_{k,d} = (1 - \mathbb{P}_d)^{k-1} \mathbb{P}_d C^{(p)}(N_d - k + 1) = \left(\prod_{j \in \mathcal{K}, j < k} (1 - \mathbb{P}_d) \right) \mathbb{P}_d C^{(p)} \left(N_d - \left(\sum_{j \in \mathcal{K}, j \leq k} 1 \right) + 1 \right) \quad (5.30)$$

However, we cannot use $\sum_{k \in \mathcal{K}, d \in \mathcal{D}} V_{k,d}$ directly in the objective function because we cannot include the penalty of unused drone and unused order into the objective function. For example, when drones $d1$ and $d2$ are considered in the system and $N_{d1} = 3$, $N_{d2} = 1$, the penalty term is $V_{1,d1} + V_{2,d1} + V_{3,d1} + V_{1,d2}$. We cannot include the penalty of order that larger than 3 for drone $d1$ and larger than 1 for drone $d2$, because drones $d1$ and $d2$ will not serve those orders. Thus, $A_{k,d}$ is introduced to copy the value of $V_{k,d}$ when drone d and order k will be used. Variable $X_{k,d}$ and the three related constraints in (5.25) - (5.27) are introduced to control the value of $A_{k,d}$.

As $X_{k,d}$ is a binary decision variable, the constraints in (5.25) and (5.26) ensure the relationship between $X_{k,d}$ and N_d . These two constraints ensure that $X_{k,d} = 1$ when $k \leq N_d$, and otherwise $X_{k,d} = 0$. For example, if $N_{d1} = 3$, then $X_{k,d1} = 1$ for $k = \{1, 2, 3\}$, and $X_{k,d1} = 0$ for $k = \{4, 5, \dots, |\mathcal{C}|\}$.

The constraint in (5.27) ensures that the pre-calculated penalty ($V_{k,d}$) is correctly stored in decision variable $A_{k,d}$. When $X_{k,d} = 1$, $A_{k,d} = V_{k,d}$. Since $X_{k,d}$ can be either 0 or 1, this constraint can result in two cases as follows:

When $X_{k,d} = 0$, the constraint in (5.27) can be ignored because Δ makes the constraint always feasible. Again, Δ denotes any large constant value, e.g. 10000.

$$V_{k,d} + 0 \leq \Delta + A_{k,d}, \forall d \in \mathcal{D}, \forall k \in \mathcal{K} \quad (5.31)$$

When $X_{k,d} = 1$, this constraint ensures that $V_{k,d} \leq A_{k,d}$ since the both sides have Δ , which are canceled each other. Moreover, the objective is to minimise Equation (5.24), and thus $A_{k,d}$ will be minimized, which makes the solution of $A_{k,d} = V_{k,d}$ when $X_{k,d} = 1$. When $X_{k,d} = 0$, the value of $A_{k,d} = 0$ because none of the constraint controls it and 0 is the smallest value of decision variable $A_{k,d}$.

$$V_{k,d} + \Delta \leq \Delta + A_{k,d}, \forall d \in \mathcal{D}, \forall k \in \mathcal{K} \quad (5.32)$$

As a result, the summation $\sum_{k=1}^{N_d} ((1 - \mathbb{P}_d)^{k-1} \mathbb{P}_d C^{(p)}(N_d - k + 1))$ in the objective function in (5.1) is replaced by the four auxiliary constraints in (5.25), (5.26), (5.27), (5.28) and $\sum_{k \in \mathcal{K}} A_{k,d}$ in the objective function in (5.24). Thus, it makes the problem linear.

5.3 Cost management

The cost management component receives the total delivery cost as an input from the package assignment component. For example, the coalition structure $\{\{p_1, p_3\}, \{p_2\}, \{p_4\}\}$ consists of coalitions $\{p_1, p_3\}$, $\{p_2\}$, and $\{p_4\}$, where only shippers p_1 and p_3 cooperate. Thus, the only coalition $\mathcal{S} = \{p_1, p_3\}$ needs to calculate the individual delivery costs for shippers p_1 and p_3 . After that, the cost management component will send the individual costs of all the shippers to the shipper cooperation component. As a result, the shipper cooperation component decides the stable coalition structure, i.e., how the shippers should form the cooperation, based on the individual delivery costs of the shippers.

With the solution of the package assignment, the delivery cost will be shared among the multiple cooperative shippers in the same coalition. Let $\mathbb{O}^{\text{ST}^*}(\mathcal{S})$ denote the total delivery cost incurred to the cooperative shippers in coalition $\mathcal{S}$. Note that $\mathbb{O}^{\text{ST}^*}(\mathcal{S})$ can be obtained from solving the package assignment problem. Let $v_p(\mathcal{S})$ denote the cost that shipper p needs to pay for the delivery when joining coalition $\mathcal{S}$. Thus, $\mathbb{O}^{\text{ST}^*}(\mathcal{S}) = \sum_{p \in \mathcal{S}} v_p(\mathcal{S})$. I use the Shapley value [109] to obtain the solution of the cost management, i.e., the delivery cost shared by the cooperative

shippers in the same coalition. The Shapley value can be expressed as follows:

$$v_p(\mathcal{S}) = \sum_{\mathcal{Q} \subseteq \mathcal{S} \setminus \{p\}} \frac{|\mathcal{Q}|!(|\mathcal{S}| - |\mathcal{Q}| - 1)!}{|\mathcal{S}|!} (\mathbb{O}^{\text{ST}*}(\mathcal{Q} \cup \{p\}) - \mathbb{O}^{\text{ST}*}(\mathcal{Q})), \quad (5.33)$$

where $|\mathcal{P}|$ represents the total number of shippers, and $|\mathcal{S}|$ represents the number of shippers in coalition $\mathcal{S}$. The four properties of Shapley value are listed as follow:

- **Efficiency** The coalition cost is equal to the summation of the individual costs, i.e., $\mathbb{O}^{\text{ST}*}(\mathcal{S}) = \sum_{p \in \mathcal{S}} v_p(\mathcal{S})$. As a result, the sum of the coalition cost will be minimised.
- **Symmetry** If shippers p and q join a coalition $\mathcal{S}$, and the total coalition cost is the same, i.e., $\mathbb{O}^{\text{ST}*}(\mathcal{S} \cup \{p\}) = \mathbb{O}^{\text{ST}*}(\mathcal{S} \cup \{q\})$, then $v_p(\mathcal{S}) = v_q(\mathcal{S})$. As such, the shared cost for shipper p and shipper q will be the same if they join coalition $\mathcal{S}$ and contribute the same.
- **Linearity** Let $\mathbb{O}^{\text{A}}$ and $\mathbb{O}^{\text{B}}$ be functions, which are similar to function $\mathbb{O}^{\text{ST}}$. When $\mathbb{O}^{\text{A}}$ and $\mathbb{O}^{\text{B}}$ are characteristic functions, $v_p(\mathcal{S}, \mathbb{O}^{\text{A}} + \mathbb{O}^{\text{B}}) = v_p(\mathcal{S}, \mathbb{O}^{\text{B}} + \mathbb{O}^{\text{A}}) = v_p(\mathcal{S}, \mathbb{O}^{\text{A}}) + v_p(\mathcal{S}, \mathbb{O}^{\text{B}})$. Note that $v_p(\mathcal{S}, \mathbb{O}^{\text{ST}})$ is referred to $v_p(\mathcal{S})$ since I only use function $\mathbb{O}^{\text{ST}}$.
- **Zero player** A shipper p does not need to pay the delivery cost if the coalition cost remains the same when shipper p joins in, i.e., $v_p(\mathcal{S}) = 0$ when $\mathbb{O}^{\text{ST}*}(\mathcal{Q}) = \mathbb{O}^{\text{ST}*}(\mathcal{Q} \cup \{p\})$ and $\mathcal{Q} \subseteq \mathcal{S}$.

5.4 Shipper Cooperation

This section presents shipper cooperation, which is the component of the BCoSDD framework. Again, the overview of the framework is presented in Section 5.1. The relationship between the three components is shown in Figure 5.1. In this section (Section 5.4), the background of shipper cooperation is described first. After that, the expected payoff and preference relations are presented. At the end of this section, two solution techniques are described.

5.4.1 Cooperation background (Coalition game)

The shippers can partner, i.e., cooperate, with each other and form a coalition for establishing a resource pool for package delivery. In this regard, all shippers need to decide to cooperate or not cooperate with other shippers to minimise their costs. Recall that $\mathcal{P}$ is a set of all shippers in the system, and $\mathcal{S}$ is a coalition, i.e., a set, of cooperative shippers establishing a resource pool. Let $\Phi = \{\phi_1, \phi_2, \dots, \phi_{|\Phi|}\}$ denote the set of all possible coalition structures, where $|\Phi|$ is the total number of all coalition structures which can be calculated from Bell number [109] with $|\mathcal{P}|$ as a parameter. In a coalition structure, there is one or more coalition, i.e., $\phi = \{\dots, \mathcal{S}_l, \dots\}$. I have $\mathcal{S}_l \cap \mathcal{S}_{l'} = \emptyset$ for all $l \neq l'$ when coalitions $\mathcal{S}_l$ and $\mathcal{S}_{l'}$ are in the same coalition structure ϕ . $\bigcup_{l \in \phi} \mathcal{S}_l = \mathcal{P}$ for every coalition structure. The coalition structure is basically a set of all coalitions that include all the shippers, i.e., $\mathcal{P} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \dots \cup \mathcal{S}_{|\phi|}$. For example, with $\mathcal{P} = \{p_1, p_2, p_3, p_4\}$ and $\phi = \{\{p_1, p_2\}, \{p_3, p_4\}\}$, this coalition structure consists of two coalitions, i.e., $\mathcal{S}_1 = \{p_1, p_2\}$ and $\mathcal{S}_2 = \{p_3, p_4\}$, where the shipper p_1 cooperates with the shipper p_2 and the shipper p_3 cooperates with the shipper p_4 .

Next, the fact that shippers can misbehave is considered when the shippers join a coalition. For example, given the solution of the package assignment for a coalition, a misbehaving shipper may not deliver packages of other shippers in the same coalition, even if they agree to cooperate. Therefore, I classify shippers into two types as follows:

- **Good** (T^G) is a shipper that delivers all assigned packages.
- **Bad** (T^B) is a shipper that does not deliver all assigned packages, and the undelivered packages belong to other shippers in the same coalition.

Each shipper has a belief probability about other shippers. Let b_{pq} denote a type that shipper p believes about shipper q . Let $P(b_{pq} = T^G)$ and $P(b_{pq} = T^B)$ denote the probabilities that shipper p believes that shipper q is a good cooperative shipper and a bad cooperative shipper, respectively. Moreover, shipper p believes that shipper q is a good cooperative shipper with probability $P(b_{pq} = T^G) = \lambda_{pq}$. Thus, $P(b_{pq} = T^B) = 1 - \lambda_{pq}$.

To address the uncertainty in the shippers' behaviour, I formulate the BCoSDD as a Bayesian coalition formation game as presented as follows:

$$G = \langle \mathcal{S}, \mathbb{T}, P, (\mu_p)_{p \in \mathcal{S}}, (\succeq_p)_{p \in \mathcal{S}} \rangle, \quad (5.34)$$

where the definitions of the parameters are given as follows.

- Let $\mathcal{S} = \{\dots, P_p, \dots\}$ denote a coalition, i.e., a set of shippers, that join the resource pool, and $\mathcal{S} \subset \mathcal{P}$. $|\mathcal{S}|$ denotes the total number of the shippers in coalition $\mathcal{S}$. Note that p and q are the indexes of the elements in set $\mathcal{S}$.
- Let $\mathbb{T} = \prod_{p \in \mathcal{S}} \mathbb{T}_p$ be the type space of all shippers in $\mathcal{S}$, where $\mathbb{T}_p = \{\mathbb{T}^G, \mathbb{T}^B\}$ denotes a set of possible types of shipper p .
- P is the probability distribution over the types of shippers, i.e., $P(b_{pq} = \mathbb{T}^B)$ and $P(b_{pq} = \mathbb{T}^G)$.
- Let μ_p denote the expected payoff, i.e., the delivery cost that shipper p needs to pay.
- Let $\succeq_p$ present the preference of shipper p .

In the following, I give the details of the expected payoff and the preference condition. Then, I present the methods to achieve the solution of the Bayesian coalition formation game, which can be referred to as the stable coalition structure. I can analyse the stable coalition structure by using the Markov transition process. I can also reach the stable coalition structure by using the merge and split algorithm.

5.4.2 Expected payoff (μ_p)

If I do not consider the uncertainty of misbehaving shippers, I can directly use the Shapley value, which is discussed in Section 5.3. Since I consider the misbehaving uncertainty, the expected payoff has to be derived. Again, μ_p denotes the expected delivery cost that shipper p needs to pay, including the penalty of the undelivered packages which is incurred from the misbehaving shipper. Let $K_p(\mathcal{S}) = \prod_{q \in \mathcal{S} \setminus \{p\}} \mathbb{T}_q$ denote the set of the possible belief combinations between shipper p and the other cooperative shippers, i.e. $q \in \mathcal{S} \setminus \{p\}$. The total number of members in $K_p(\mathcal{S})$ is

$2^{(|S|-1)}$. For example, if $\mathcal{S} = \{p_1, p_2, p_3\}$ and $p = p_1$, $K_1(\mathcal{S}) = \{k_1, k_2, \dots, k_{2^{|S|-1}}\} = \mathbb{T}_2 \times \mathbb{T}_3 = \{k_1 = (b_{12} = \mathbb{T}^G, b_{13} = \mathbb{T}^G), k_2 = (b_{12} = \mathbb{T}^G, b_{13} = \mathbb{T}^B), k_3 = (b_{12} = \mathbb{T}^B, b_{13} = \mathbb{T}^G), k_4 = (b_{12} = \mathbb{T}^B, b_{13} = \mathbb{T}^B)\}$. The expected delivery cost of shipper p can be calculated from

$$\mu_p(\mathcal{S}) = v_p(\mathcal{S}) + \sum_{k \in K_p(\mathcal{S})} \sum_{q \in S \setminus \{p\}} \Omega_{kp} \Lambda_{kpq}, \quad (5.35)$$

where Ω_{kp} is the probability that the belief combination k happens, and Λ_{kpq} denotes the incurred cost when the belief combination k happens, and shipper q does not deliver packages for shipper p . Again, $v_p(\mathcal{S})$ can be obtained from the cost management in Section 5.3. Ω_{kp} is obtained from

$$\Omega_{kp} = \prod_{q \in S \setminus \{p\}} P(b_{pq} = t), \quad (5.36)$$

where t is the type taken from space $\mathbb{T}$, and Λ_{kpq} is obtained from

$$\Lambda_{kpq} = \begin{cases} 0, & \text{if } b_{pq} = \mathbb{T}^G, \\ \theta_{pq} C^{(p)}(1 - \lambda_{pq}), & \text{if } b_{pq} = \mathbb{T}^B. \end{cases} \quad (5.37)$$

θ_{pq} denotes the total number of the packages that shipper p transfers to shipper q , i.e., $\theta_{pq} = \sum_{i \in \mathcal{C}} M_{p,q}^*$, where $M_{p,q}^*$ can be obtained from the package assignment in Section 5.2.

5.4.3 Preference relations

I adopt the preference relations from [145]. Again, let $\succeq_p$ present the preference of shipper p . For example, $(\mathcal{S}_1, \phi_m) \succeq_p (\mathcal{S}_2, \phi_n)$ means that shipper p prefers to join coalition $\mathcal{S}_1$ from coalition structure ϕ_m over to join coalition $\mathcal{S}_2$ from coalition structure ϕ_n . The preference relation of a shipper can be calculated from

$$(\mathcal{S}_1, \phi_m) \succeq_p (\mathcal{S}_2, \phi_n) \longleftrightarrow \nu_p(\mathcal{S}_1, \phi_m) \geq \nu_p(\mathcal{S}_2, \phi_n), \quad (5.38)$$

FIGURE 5.3: The state transition diagram of three shippers.

where $p \in \mathcal{S}_1$, $p \in \mathcal{S}_2$, and $\mathcal{S}_1$ and $\mathcal{S}_2$ are any two coalitions. The preference function is represented as ν_p and is defined as follows:

$$\nu_p(\mathcal{S}, \phi) = \begin{cases} \mu_p(\mathcal{S}, \phi), & \text{if } \mu_q(\mathcal{S}, \phi) \leq \mu_q(\mathcal{S} \setminus \{q\}, \phi) \\ & \forall q \in \mathcal{S} \setminus \{q\} \text{ and } |\mathcal{S}| \neq 1 \\ 0 & \text{otherwise,} \end{cases} \quad (5.39)$$

This function requires three inputs, i.e., (i) a main shipper p , (ii) coalition $\mathcal{S}$, and (iii) the coalition structure of coalition $\mathcal{S}$, which is ϕ . From (5.39), shipper p will switch from the current coalition $\mathcal{S}_m$ to a another coalition $\mathcal{S}_n$ when shipper p prefers the new coalition $\mathcal{S}_n$ over coalition $\mathcal{S}_m$, i.e., $(\mathcal{S}_n \cup \{p\}, \phi_n) \succeq_p (\mathcal{S}_m, \phi)$, where $\mathcal{S}_m$ and $\mathcal{S}_n \in \phi$. As a result, the new coalition structure is $\phi_n = \{\phi \setminus \{\mathcal{S}_m, \mathcal{S}_n\}\} \cup \{\mathcal{S}_m \setminus \{p\}, \mathcal{S}_n \cup \{p\}\}$.

5.4.4 Solution technique: merge and split algorithm

According to the coalition structure switching, one shipper can join or leave a coalition at a time. Therefore, the merge and split algorithm is adopted from [145] as presented in Algorithm 2. The algorithm is composed of four major steps, i.e., neighbouring coalition search (Line 3), the calculation of Shapley value (Line 8 to Line 9), the calculation of Bayesian cost (Line 12 to Line 17), and stable state checking (Line 18 to Line 21). The algorithm terminates when the current coalition structure ϕ_m cannot change to any neighbouring coalition structure.

Algorithm 2: Merge and split algorithm for cooperation management

Input: Set $\mathcal{P}$ and all the coalition cost $V(\mathcal{S})$ obtained by solving the package assignment optimisation problem

Output: A stable coalition structure, the individual cost for each shipper, the drone delivery planning

Initialisation : All shippers do not cooperate

```

1: while  $\phi_m$  changes do
2:    $\phi_n =$  The list of  $\phi_m$  neighbouring coalition structures, which can be found
      by using the neighbouring coalition discovery algorithm [145], [13].
3:   for every  $b \in \phi_n$  do
4:     for every  $p \in \mathcal{P}$  do
5:        $\mathcal{C}_o$  and  $\mathcal{C}_b$  are the coalitions that  $p$  belongs to coalition structure  $\phi_m$ 
      and  $\phi_n$ , respectively.
6:       for every  $S \in \phi_m$  do
7:         Obtain  $\mathbb{O}^{ST*}(\mathcal{S})$ ,  $\forall S \in \phi_m$  //call the coalition cost, which can be
          obtained by solving the package assignment optimisation problem of
          coalition  $S$ 
           $v_p(S) \leftarrow \mathbb{O}^{ST*}(\mathcal{S})$  //solve for the Shapley Value
8:       end for
9:       for every  $S \in \phi_n$  do
10:        Obtain  $\mathbb{O}^{ST*}(\mathcal{S})$ ,  $\forall S \in \phi_n$  //call the coalition cost, which can be
          obtained by solving the package assignment optimisation problem of
          coalition  $S$ 
           $v_p(S) \leftarrow \mathbb{O}^{ST*}(\mathcal{S})$  //solve for the Shapley Value
11:       end for
12:       if  $p$  has visited  $\mathcal{C}_b$  before and  $|\mathcal{C}_b| > 1$  then
13:          $v_p(\mathcal{C}_b, \phi_n) = 0$ 
14:       else
15:          $v_p(\mathcal{C}_b, \phi_n)$  is calculated from (5.39)
16:       end if
17:       Calculate  $v_p(\mathcal{C}_o, \phi_m)$  from (5.39)
18:       if  $v_p(\mathcal{C}_o, \phi_m) \geq v_p(\mathcal{C}_b, \phi_n)$  then
19:          $\phi_m = \phi_n$ ; // the coalition structure has changed break; // break to
          the while loop (Line 2)
20:       end if
21:     end for
22:   end for
23: end while

```

5.4.5 Solution technique: markov transition process

Let a set of coalition structures, i.e., $\Phi = \{\phi_1, \phi_2, \dots, \phi_{|\Phi|}\}$, be the state space of the Markov transition process. Let ρ_{ϕ_m, ϕ_n} denote a transition probability when transiting from state ϕ_m to state ϕ_n , which can be referred to as coalition structure ϕ_m is changed to coalition structure ϕ_n . Based on the preference relations in Section 5.4.3, only one shipper can change from coalition structure ϕ_m to coalition structure ϕ_n at a time. Figure 5.3 shows the example of the state transition diagram when there are three shippers. Let $\mathcal{H}_\phi$ denote the set of the neighbouring coalition structures of coalition structure ϕ , which only one shipper can change a coalition at a time. For example, if $\phi_m = \{\{p_1\}, \{p_2\}, \{p_3\}\}$, $\mathcal{H}_{\phi_m} = \{\{\{p_1, p_2\}, \{p_3\}\}, \{\{p_1, p_3\}, \{p_2\}\}, \{\{p_2, p_3\}, \{p_1\}\}\}$. Let $\mathbb{B}_{\phi_m, \phi_n}$ denote the probability that coalition structure ϕ_m will change to coalition structure ϕ_n , which can be calculated from

$$\mathbb{B}_{\phi_m, \phi_n} = \begin{cases} 1 - \sum_{\phi \in \Phi} \mathbb{B}_{\phi_m, \phi}, & \text{if } \phi_m = \phi_n, \\ 0, & \text{else if } \phi_n \notin \mathcal{H}_{\phi_m}, \\ \varepsilon \beta_{\phi_m, \phi_n}, & \text{else if} \\ & \nu_b(S_m, \phi_m) < \nu_b(S_n, \phi_n), \\ & \text{for all } b \in \mathcal{B}_{\phi_m, \phi_n} \\ (1 - \varepsilon) \beta_{\phi_m, \phi_n}, & \text{otherwise,} \end{cases} \quad (5.40)$$

$$\beta_{\phi_m, \phi_n} = \alpha^{|\mathcal{B}_{\phi_m, \phi_n}|} (1 - \alpha)^{|\mathcal{P}| - |\mathcal{B}_{\phi_m, \phi_n}|} \quad (5.41)$$

where $\mathcal{B}_{\phi_m, \phi_n}$ represents the set of customers that are involved to the transition from coalition structure ϕ_m to coalition structure ϕ_n , and $|\mathcal{B}_{\phi_m, \phi_n}|$ denotes the total number of the shippers in set $\mathcal{B}_{\phi_m, \phi_n}$. Note that α denotes the probability that a shipper changes its coalition.

Given transition probability matrix Q , which is expressed as follows:

$$Q = \begin{bmatrix} \mathbb{B}_{1,1} & \mathbb{B}_{1,2} & \dots & \mathbb{B}_{1,|\Phi|} \\ \mathbb{B}_{2,1} & \mathbb{B}_{2,2} & \dots & \mathbb{B}_{2,|\Phi|} \\ \vdots & \vdots & \ddots & \\ \mathbb{B}_{|\Phi|,1} & \mathbb{B}_{|\Phi|,2} & & \mathbb{B}_{|\Phi|,|\Phi|} \end{bmatrix}, \quad (5.42)$$

I can obtain the stationary probability vector by solving

$$\vec{\Pi}^\top Q = \vec{\Pi}^\top, \quad \vec{\Pi}^\top \vec{1} = 1, \quad (5.43)$$

where $\vec{\Pi} = [\pi_1, \pi_2, \dots, \pi_{|\Phi|}]^\top$ denotes the stationary probability vector, and π_ϕ is the probability that coalition structure ϕ will be formed.

5.5 Belief update mechanism

As discussed in the previous section, I can use the merge and split algorithm to obtain a stable coalition structure in the static Bayesian coalition formation game. In this section, I extend the static Bayesian coalition formation game to the dynamic one that allows a shipper to observe behaviours and update the belief about the other cooperative shippers in the same coalition continuously. Therefore, the dynamic Bayesian coalition formation game can be used even when the shippers do not have prior knowledge about the other shippers. In particular, a shipper observes the behaviour of the cooperative shippers whether all the packages are successfully delivered by them or not. The belief probability, i.e., $P(b_{p,q} = T^G)$ and $P(b_{p,q} = T^B)$, is updated based on the observation.

I propose the belief update mechanism based on the Bayes' theorem [66] and the algorithm in [68]. There are three possible conditions involved in the belief update mechanism of shipper p about shipper q , namely: (i) $\Omega^{(F)}$ is the condition that shipper q 's delivery is not successful due to technical issues, (ii) $\Omega^{(G)}$ is the condition that shipper q is a cooperative shipper., and (iii) $\Omega^{(B)}$ is the condition that shipper q is a bad cooperative shipper, i.e., its real type is **Bad**. Let $\Omega = T$ represent that the condition Ω is true, and $\Omega = F$ otherwise, where $\Omega \in \{\Omega^{(F)}, \Omega^{(G)}, \Omega^{(B)}\}$. Here, I have $P_{p,q}(\Omega^{(B)} = T) = P_{p,q}(\Omega^{(G)} = F) = 1 - P_{p,q}(\Omega^{(G)} = T)$. The probability that the condition $\Omega^{(G)}$ is true is equal to the belief probability, i.e., $P_{p,q}(\Omega^{(G)} =$

FIGURE 5.4: Tree diagram of the belief update mechanism of shipper p about shipper q .

$P(b_{p,q} = T^G) = \lambda_{p,q}$. Figure 5.4 presents the tree diagram of the belief update mechanism with the three conditions and their corresponding probabilities. Let ϵ denote an error probability that the delivery by the good cooperative shipper is unsuccessful.

Likewise, Table 5.1 shows all the joint probabilities. There are eight cases, but only three cases can happen. The cases ω_1 , ω_4 , ω_5 and ω_8 will not happen because $\Omega^{(G)}$ and $\Omega^{(B)}$ cannot be true at the same time, and ω_3 will not happen because the bad cooperative shipper will not deliver the package. From Table 5.1, I can show that

$$P_{p,q}(\Omega^{(F)} = T | \Omega^{(G)} = T) + P_{p,q}(\Omega^{(F)} = F | \Omega^{(G)} = T) + P_{p,q}(\Omega^{(B)} = T) = 1, \quad (5.44)$$

where

$$P_{p,q}(\Omega^{(F)} = T | \Omega^{(G)} = T) = \lambda_{p,q}\epsilon, \quad (5.45)$$

$$P_{p,q}(\Omega^{(F)} = F | \Omega^{(G)} = T) = \lambda_{p,q}(1 - \epsilon), \quad (5.46)$$

$$P_{p,q}(\Omega^{(B)} = T) = 1 - \lambda_{p,q}. \quad (5.47)$$

Consequently, the probabilities that a package of shipper p is delivered successfully and unsuccessfully by shipper q , i.e., $P_{p,q}(S)$ and $P_{p,q}(U)$, can be calculated from

$$P_{p,q}(S) = P_{p,q}(\Omega^{(F)} = F | \Omega^{(G)} = T) = \frac{\theta'_{p,q}}{\theta_{p,q}} \quad (5.48)$$

$$\begin{aligned} P_{p,q}(U) &= P_{p,q}(\Omega^{(F)} = T | \Omega^{(G)} = T) + P_{p,q}(\Omega^{(B)} = T) \\ &= 1 - \frac{\theta'_{p,q}}{\theta_{p,q}}, \end{aligned} \quad (5.49)$$

where $\theta'_{p,q}$ denotes the exact number of the packages that shipper q successfully delivers for shipper p . Again, $\theta_{p,q}$ represents the total number of the packages that shipper p transfers to shipper q , and shipper q needs to deliver them. From (5.48)

TABLE 5.1: Truth table to the joint probabilities of the belief update mechanism.

Scenario	$\Omega^{(F)}$	$\Omega^{(G)}$	$\Omega^{(B)}$	Joint Probability
ω_1	T	T	T	0
ω_2	T	T	F	$\lambda_{p,q}\epsilon$
ω_3	T	F	T	0
ω_4	T	F	F	0
ω_5	F	T	T	0
ω_6	F	T	F	$\lambda_{p,q}(1 - \epsilon)$
ω_7	F	F	T	$1 - \lambda_{p,q}$
ω_8	F	F	F	0

and (5.49), I can obtain the new belief probability $\lambda_{p,q}^{t+1}$ in iteration t based on the observation as follows:

$$\lambda_{p,q}^{t+1} = \begin{cases} \frac{\theta'_{p,q}}{\theta_{p,q}(1 - \epsilon)}, & \text{if } \frac{\theta'_{p,q}}{\theta_{p,q}} < 1 - \epsilon \\ 1, & \text{otherwise.} \end{cases} \quad (5.50)$$

However, the new belief probability $\lambda_{p,q}$ from (5.50) is independent of the past as the probability is based on only current observation. Therefore, the weight adjustment from the exponential moving average (EMA) method [97] can be used to update the belief probability as follows:

$$\lambda_{p,q}^{t+1} = w_1 \lambda_{p,q}^t + w_2 \lambda_{p,q}^{t+1} \quad (5.51)$$

where $\lambda_{p,q}^{t+1}$ on the right hand side is obtained from (5.50). The constants w_1 and w_2 are adjustable weight parameters, for $w_1 + w_2 = 1$ and $0 \leq w_1, w_2 \leq 1$.

Algorithm 3 shows all steps of dynamic Bayesian coalition formation. The belief update mechanism, which is based on (5.44) to (5.51) and the merge and split algorithm are used in Algorithm 3. When the shippers perform the package delivery by following the planning solution for multiple times, the belief probabilities converge to constants, and thus the algorithm also converges to the same stable coalition structure as achieved when all the types of shippers are exactly known.

Algorithm 3: Dynamic Bayesian Coalition Formation**Output:** Drone delivery planning solution

```

1: repeat
2:   Perform Algorithm 2
3:   Perform package delivery based on the solution of Algorithm 2
4:   Observe whether the packages are successfully delivered or not, i.e.,  $\theta'_{p,q}$ .
5:   for every  $p \in \mathcal{P}$  do
6:     for every  $q \in \mathcal{P}$  do
7:       if shipper  $p$  transfers packages to shipper  $q$  then
8:         Update the belief probability as in (5.51)
9:       else
10:         $\lambda_{p,q}^{t+1} \leftarrow \lambda_{p,q}^t$ 
11:       end if
12:     end for
13:   end for
14:    $t \leftarrow t + 1$ 
15: until Terminates

```

5.6 Performance evaluation

In this section, I first describe the experimental setup. Next, I discuss the experimental results to demonstrate the performance of the BCoSDD framework.

5.6.1 Experiment setup

I consider four shippers, i.e., $\mathcal{P} = \{p_1, p_2, p_3, p_4\}$. Each shipper $p \in \mathcal{P}$ has one drone, and their drones are identical. The initial cost of the drones is set as $C^{(i)} = S\$100$. The limits of the drones are set as $f_d = 5$ kilograms, $e_d = 10$ kilometres, $l_d = 150$ kilometres, and $h_d = 8$ hours. I use the average flying speed of drones which is $s_d = 30$ kilometres per hour in the experiments. The serving time is set as $t_i = 15$ minutes for all $i \in \mathcal{C}$. All the customer packages in the experiments weight less than 5 kilograms, i.e., $a_i \leq 5$ kilograms. When the shippers cooperate, they share their customers and drones in the resource pool. The transferring cost from the original depot to the new depot is set as $C^{(t)} = S\$30$ per transfer. The outsourcing cost is set as $C_i^{(c)} = S\$16$ based on the speedpost service provided by Singpost company [128]. Similarly, the penalty of breakdown is set as $C_i^{(p)} = S\$16$. Therefore, the shippers may handle the unsuccessful delivery packages by outsourcing them to the carrier. The probability that shipper p believes that

(A) The synthesised Solomon Benchmark data.

(B) A real data from logistics industry in Singapore.

FIGURE 5.5: The locations of customers and depots.

shipper q are a good cooperative shipper is set as $P(b_{pq} = T^G) = 0.9$ for all $p, q \in \mathcal{P}$.

I evaluate the BCoSDD framework with two datasets of customers' locations, i.e., the Solomon benchmark suites [93] and a real dataset from logistics industry in Singapore. The Solomon benchmark file C101 is synthesised to have four depots and 60 customers, i.e., $|c| = 60$. Alternatively, 100 customers, i.e., $|c| = 100$, are considered for the real dataset experiments. The locations of depots and customers in the two datasets are shown in Figure 5.5a. The number of customers is shared equally among the shippers. I set $C_1 = \{c_1, c_5, \dots, c_{c'-3}\}$ to be the customers of shipper p_1 , $C_2 = \{c_2, c_6, \dots, c_{c'-2}\}$ to be the customers of shipper p_2 , $C_3 = \{c_3, c_7, \dots, c_{c'-1}\}$ to be the customers of shipper p_3 , and $C_4 = \{c_4, c_8, \dots, c_{c'}\}$ to be the customers of shipper p_4 .

I implement the optimisation model of package assignment as a linear programming in GAMS [22] and, it is solved by a commercial solver CPLEX [22].

5.6.2 Analysis of cooperation management

The initial cost of drones is set as $C_d^{(i)} = S\$0$ and $C_d^{(i)} = S\$90$, where $C_d^{(i)} = S\$0$ is when a shipper has its own drone and does not need to pay for the initial cost. I consider the traditional approach which is referred to as deterministic approach, and the Bayesian approach, i.e., the BCoSDD framework. The deterministic approach considers the case that all shippers are the good cooperative shippers, and they always deliver all packages successfully. However, the Bayesian approach considers the case that all shippers believe that other cooperative shippers may fail to deliver packages with probability $P(b_{pq} = T^B) = 1 - P(b_{pq} = T^G) = 0.1$ for all $p, q \in \mathcal{P}$. Table 5.2 shows the individual delivery cost for all the coalition structures, i.e., Φ_1 to Φ_{15} . The stationary probabilities obtained from the Markov model are presented in Figure 5.6a. The selected coalition structures of the merge and split algorithm are shown in Figure 5.6b, where the last selection is the stable coalition structure.

When I set $C_d^{(i)} = S\$0$, and there are three stable coalition structures in the deterministic and Bayesian approaches, i.e., Φ_{11} , Φ_{14} , and Φ_{15} . The coalition structures Φ_{14} and Φ_{15} have the highest probabilities in the deterministic approach and the

Bayesian approach, respectively. Identically, the merge and split algorithm achieves Φ_{14} and Φ_{15} as the stable coalition structures for the deterministic approach and the Bayesian approach, respectively. Since both shippers p_1 and p_4 want to cooperate with shipper p_3 , and shipper p_3 can choose to cooperate such that its cost is minimised. As a result, shipper p_3 cooperates with shipper p_4 , and the resulting coalition structure is Φ_7 for the deterministic approach. By contrast, shipper p_3 cooperates with shipper p_1 , and the resulting coalition structure is Φ_3 for the Bayesian approach. For the deterministic approach, shippers p_3 and p_4 allow shipper p_2 to join the coalition in the seventh iteration, and the corresponding coalition structure is Φ_{14} to lower the cost. The stable coalition structure is Φ_{14} as shipper p_4 does not allow shipper p_1 to join the coalition because the shipper p_4 will pay a higher cost if coalition structure Φ_{15} is reached. On the other hand, coalition structure Φ_{15} is stable for the Bayesian approach because all the shippers achieve the lowest cost when this coalition structure is reached.

When I set $C_d^{(i)} = S\$90$, Φ_{15} is the stable coalition structure from both the Markov model and the merge and split algorithm for the deterministic approach. However, there is no stable coalition structure for the Bayesian approach since all the stationary probabilities are less than 0.13 and the merge and split algorithm terminates with the initial coalition structure. As such, the shippers should not cooperate with each other as the initial cost $C_d^{(i)} = S\$90$ is high, and hence the cooperation results in higher cost than outsourcing the package delivery to the carrier.

I have shown that the merge and split algorithm can achieve the same solution as that from the Markov model. While the Markov model requires the Shapley values of all the coalition structures before being solved, the merge and split algorithm needs only the Shapley values of the neighbouring coalition structures when changing the selected coalition structure. For example, the merge and split algorithm does not require the Shapley values of coalition structures Φ_{11} and Φ_{12} in the experiment for the deterministic approach with $C_d^{(i)} = S\$0$.

TABLE 5.2: Shapley value and coalition structures

		Deterministic (Initial Cost = 0)				Bayesian (Initial Cost = 0)				Deterministic (Initial Cost = 90)				Bayesian (Initial Cost = 90)			
Coalition Structure		P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4
$\Phi_1 = \{\{p_1\}, \{p_2\}, \{p_3\}, \{p_4\}\}$		174.74	211.01	186.13	154.33	174.74	211.01	186.13	154.33	240.00	240.00	240.00	240.00	240.00	240.00	240.00	240.00
$\Phi_2 = \{\{p_1, p_2\}, \{p_3\}, \{p_4\}\}$		174.75	211.02	186.13	154.33	174.75	211.02	186.13	154.33	240.00	240.00	240.00	240.00	240.00	240.00	240.00	240.00
$\Phi_3 = \{\{p_1, p_3\}, \{p_2\}, \{p_4\}\}$		158.05	211.01	169.44	154.33	166.05	211.01	175.84	154.33	240.00	240.00	240.00	240.00	240.00	240.00	240.00	240.00
$\Phi_4 = \{\{p_1, p_4\}, \{p_2\}, \{p_3\}\}$		174.74	211.01	186.13	154.33	174.74	211.01	186.13	154.33	240.00	240.00	240.00	240.00	240.00	240.00	240.00	240.00
$\Phi_5 = \{\{p_2, p_3\}, \{p_1\}, \{p_4\}\}$		174.74	211.01	186.13	154.33	174.74	211.01	186.13	154.33	240.00	240.00	240.00	240.00	240.00	240.00	240.00	240.00
$\Phi_6 = \{\{p_2, p_4\}, \{p_1\}, \{p_3\}\}$		174.74	202.38	186.13	145.70	174.74	211.98	186.13	148.90	240.00	240.00	240.00	240.00	240.00	240.00	240.00	240.00
$\Phi_7 = \{\{p_3, p_4\}, \{p_1\}, \{p_2\}\}$		174.74	211.01	165.49	133.69	174.74	211.01	178.29	140.09	240.00	240.00	236.87	236.87	240.00	240.00	251.27	236.87
$\Phi_8 = \{\{p_1, p_2\}, \{p_3, p_4\}\}$		174.75	211.02	165.49	133.69	174.75	211.02	178.29	140.09	240.00	240.00	236.87	236.87	240.00	240.00	251.27	236.87
$\Phi_9 = \{\{p_1, p_3\}, \{p_2, p_4\}\}$		158.05	202.38	169.44	145.70	166.05	211.98	175.84	148.90	240.00	240.00	240.00	240.00	240.00	240.00	240.00	240.00
$\Phi_{10} = \{\{p_1, p_4\}, \{p_2, p_3\}\}$		174.74	211.01	186.13	154.33	174.74	211.01	186.13	154.33	240.00	240.00	240.00	240.00	240.00	240.00	240.00	240.00
$\Phi_{11} = \{\{p_1, p_2, p_3\}, \{p_4\}\}$		137.19	190.15	148.57	154.33	148.39	198.15	158.17	154.33	240.00	240.00	240.00	240.00	240.00	240.00	240.00	240.00
$\Phi_{12} = \{\{p_1, p_2, p_4\}, \{p_3\}\}$		167.69	195.33	186.13	138.65	174.09	204.93	186.13	148.24	240.00	240.00	240.00	240.00	240.00	240.00	240.00	240.00
$\Phi_{13} = \{\{p_1, p_3, p_4\}, \{p_2\}\}$		152.77	211.01	143.52	128.41	163.97	211.01	159.52	141.21	229.97	240.00	226.84	226.84	244.37	240.00	239.64	233.24
$\Phi_{14} = \{\{p_2, p_3, p_4\}, \{p_1\}\}$		174.74	184.64	147.75	107.31	174.74	199.04	162.15	116.91	240.00	234.65	231.52	231.52	240.00	249.05	242.71	237.92
$\Phi_{15} = \{\{p_1, p_2, p_3, p_4\}\}$		138.61	170.48	118.67	108.74	153.01	184.88	136.27	124.74	227.20	231.88	218.71	218.71	241.60	246.28	236.31	234.71

FIGURE 5.6: Analysis of the stable coalition formation

5.6.3 Impact of the optimisation

To understand the results in the cooperation management more deeply, I thus evaluate only the package assignment of the BCoSDD framework. I conduct four experiments including (i) varying the penalty $C_i^{(p)}$ and the outsourcing cost $C_i^{(c)}$, (ii) varying the transferring cost $C_p^{(t)}$, (iii) varying the breakdown probability $\mathbb{P}_d$, and (iv) varying the initial cost $C_d^{(i)}$. Figures 5.7(a), (e), (g), and (j) show the costs of the coalition, when all the shippers cooperate, i.e., the coalition structure Φ_{15} . Similarly, Figures 5.7(b), (e), (h), and (k) show the numbers of customers which are served by the drones, served by the outsourcing carrier, and whose packages are transferred to another depot under the coalition structure Φ_{15} . Figures 5.7(c), (f), (i), and (l) show the impact of the parameters to the stable coalitions.

When the penalty and the outsourcing cost are set such that $C^{(p)} = C_i^{(c)}$ and are varied, the shippers outsource all packages to the carrier when the outsourcing cost is cheap, i.e. $C_i^{(c)} \leq S\$14$. As a result, the shippers do not cooperate. If all the shippers are cooperate (Φ_{15}), three drones are used to serve 44 customers for $C_i^{(c)} = S\$16$, and four drones are used to serve 51 customers for $S\$16 \leq C_i^{(c)} \leq S\20 . The remaining customers are served by the carrier. Note that the drone cannot serve 9 customers because they are outside the serving area as indicated in Figure 5.5a.

All the shippers will cooperate when $C_i^{(c)} \geq S\$18$ for the deterministic approach and $C_i^{(c)} = S\$20$ in the Bayesian approach. However, the experiment of varying transferring cost ($C_p^{(t)}$) has a different result. Specifically, all the shippers cooperate (Φ_{15}) when the transferring cost is cheap ($C_p^{(t)} = S\$0$), and they do not cooperate when the transferring cost is high. If all of the shippers cooperate, i.e., Φ_{15} , three drones are used to serve 44 customers, and 34 packages are transferred when $C_p^{(t)} \leq S\$30$. Moreover, when all the shippers cooperate (Φ_{15}), the experiments of varying the breakdown probability and varying the initial cost share a similar trend. In particular, four drones are used when the breakdown probability is small ($\mathbb{P}_d \leq 0.025$) and when the initial cost is cheap ($C_d^{(i)} \leq S\$90$). After the parameters increase to a certain point, i.e., $\mathbb{P}_d = 0.0375$ and $C_d^{(i)} = S\$105$, the number of allocated drones decreases to three. After increasing the parameters further until the drone delivery cost is higher than the outsourcing cost, no drone is used when $\mathbb{P}_d \geq 0.0625$ and $C_d^{(i)} \geq S\$120$ for the experiment of varying breakdown probability and varying the initial cost, respectively.

Based on Figure 5.7c, I can conclude that the stable coalition structure is affected by the system parameters. From the experiments, the deterministic approach tends to let the shippers join a coalition easier and leave the coalition harder than the Bayesian approach. Again, the deterministic approach considers the case that all shippers always deliver all packages successfully. For example, when the outsourcing cost is high, the deterministic approach lets the shippers form coalition structure Φ_{15} since $C_i^{(c)} = S\$18$ while the Bayesian approach lets them deliver individually and form the same coalition later at $C_i^{(c)} = S\$20$. In contrast, the Bayesian approach lets shippers p_1 and p_2 leave the coalition Φ_{15} immediately after the transferring cost increases to $C_p^{(t)} = S\$10$ while the deterministic approach lets all the shippers cooperate Φ_{15} until $C_p^{(t)} = S\$20$. Note that I have explained the impact of the initial cost in Section 5.6.2, especially the reason that the stable coalition structure is Φ_{14} when the initial cost is small, i.e., $C^{(i)} = S\$0$.

5.6.4 Simulation

I conduct the simulation experiments to validate the numerical results in which breakdown events are simulated. The simulation code is written in Matlab [23]. The parameters are set as follows. The outsourcing cost is set as $C_i^{(c)} = S\$16$. Four shippers, i.e., $\mathcal{P} = \{p_1, p_2, p_3, p_4\}$, are considered. Each shipper $p \in \mathcal{P}$ has one drone, and their drones are identical. In the first experiment, the breakdown probability is varied. The cooperation is not considered in the first experiment. In the second experiment, the breakdown probability is set as 0.05. The probability that shipper p believes that shipper q is a good cooperative shipper is set as $P(b_{pq} = T^G) = 0.9$ for all $p, q \in \mathcal{P}$. The other parameters are set based on the setting in Section 5.6.1.

- In the first experiment, I focus on the package assignment with two cases. In the first case, the package assignment does not consider the breakdown when obtaining the solution, and in the second case, it considers the breakdown. The first and second cases are referred to as the deterministic drone delivery (DDD) and the stochastic drone delivery (SDD), shown in Figure 5.8, respectively.

FIGURE 5.7: Impact of Optimisation

- In the second experiment, I focus on the package assignment, cost sharing, and cooperation management. I compare the total cost between our proposed approach, i.e., BCoSDD framework, and four baseline approaches in Figure 5.9 and Figure 5.10. The four baseline approaches are listed as follows.
 - **Deterministic drone delivery (DDD)** uses deterministic optimisation approach with none of the shippers cooperates
 - **Stochastic drone delivery (SDD)** uses the stochastic optimisation approach with none of the shippers cooperates
 - **Shipper cooperative in deterministic drone delivery (CoDDD)** uses the deterministic optimisation approach with deterministic cooperative game (cooperate without considering misbehavior)
 - **Shipper cooperative in stochastic drone delivery (CoSDD)** uses the stochastic optimisation approach with deterministic cooperative game (cooperate without considering misbehavior)

Nonetheless, in all the approaches, drone can breakdown and the cooperative shippers can misbehave.

I experiment the SDD approach with only one shipper, i.e., p_3 . From Figure 5.8, the total cost achieved from the SDD approach is lower than or equal to that of the DDD approach. When the breakdown probability and penalty cost increase, the SDD approach tends to assign packages to the carrier more frequently than the DDD approach since the DDD approach does not consider the breakdown. As such, the SDD approach can avoid the penalty cost better than the DDD approach. Therefore, the SDD approach achieves the lower total cost than that of the DDD approach when the breakdown probability is $0.15 \leq \mathbb{P}_d \leq 0.2$ and when the breakdown penalty is $S\$26 \leq C^{(p)} \leq S\30 .

Furthermore, Figure 5.9 presents the individual cost that each shipper needs to pay when using different approaches. In this experiment, I set the initial cost of drones to S\$100. The cost of each shipper for each coalition structure can be found in Table 5.3. The BCoSDD approach achieves the stable coalition structure Φ_6 , which shippers p_2 and p_4 cooperate. However, the CoDDD and CoSDD approaches achieve the stable coalition structures Φ_{15} and Φ_{13} , respectively. Based on the simulation results, the solution from the CoDDD approach favours shipper p_2 , but

FIGURE 5.8: Simulation results of deterministic (DDD) and stochastic (SDD) approaches of shipper P_3 with one drone.

the solution is not fair to the other shippers as they will pay a higher delivery cost than that when they do not cooperate. Similarly, the solution from the CoSDD approach favors shipper p_3 .

In Figure 5.10, I assume that the shippers own drones and do not need to pay for the initial cost, i.e., rental fee. The cost of each shipper for each coalition structure are shown in Table 5.4. From Table 5.4, the CoDDD and CoSDD approaches achieve the stable coalition structure Φ_{13} . CoDDD approach predicts that shipper p_3 can earn some revenue by cooperating with the other shippers (the value is negative). However, the simulation result based on CoDDD solution (in Figure 5.10) shows that shipper p_3 will need to pay S\$9.5 instead of earning due to drone breakdown and misbehaving shippers. The CoDDD solution does not favor shippers p_1 and p_4 as they need to pay more than that of not cooperating. Similarly, the CoSDD solution is also unfair to shipper p_4 . The BCoSDD approach achieves different stable coalition structure, which is Φ_3 , where all the shippers in the stable coalition structure, i.e., shippers p_1 and p_3 , pay lower delivery cost than that of performing the delivery individually.

Apparently, from the simulation experiments, the distributed cost for each shipper from the BCoSDD approach is always reasonable and the lowest cost compared with the DDD, CoDDD, SDD, and CoSDD approaches.

FIGURE 5.9: The comparison of simulation results when the initial cost of drone = S\$100

FIGURE 5.10: The comparison of simulation results when shippers own the drones (the initial cost of drone = S\$0)

TABLE 5.3: Shapley value of the different approaches for the decisions in the simulation experiment (when the initial cost of drones is set as S\$100))

Coalition Structure	CoDDD				CoSDD				BCoSDD			
	P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4
$\Phi_1 = \{\{p_1\}, \{p_2\}, \{p_3\}, \{p_4\}\}$	400.00	317.94	129.33	333.16	400.00	370.21	298.09	378.08	400.00	370.21	298.09	378.08
$\Phi_2 = \{\{p_1, p_2\}, \{p_3\}, \{p_4\}\}$	373.86	291.81	129.33	333.16	400.00	370.21	298.09	378.08	400.00	370.21	298.09	378.08
$\Phi_3 = \{\{p_1, p_3\}, \{p_2\}, \{p_4\}\}$	301.18	317.94	30.51	333.16	379.91	370.21	278.00	378.08	416.71	370.21	278.00	378.08
$\Phi_4 = \{\{p_1, p_4\}, \{p_2\}, \{p_3\}\}$	369.03	317.94	129.33	302.19	400.00	370.21	298.09	378.08	400.00	370.21	298.09	378.08
$\Phi_5 = \{\{p_2, p_3\}, \{p_1\}, \{p_4\}\}$	400.00	246.09	57.48	333.16	400.00	362.39	290.27	378.08	400.00	402.39	290.27	378.08
$\Phi_6 = \{\{p_2, p_4\}, \{p_1\}, \{p_3\}\}$	400.00	233.68	129.33	248.90	400.00	352.07	298.09	359.94	400.00	368.07	298.09	375.94
$\Phi_7 = \{\{p_3, p_4\}, \{p_1\}, \{p_2\}\}$	400.00	317.94	56.87	260.70	400.00	370.21	287.55	367.54	400.00	370.21	287.55	405.94
$\Phi_8 = \{\{p_1, p_2\}, \{p_3, p_4\}\}$	373.86	291.81	56.87	260.70	400.00	370.21	287.55	367.54	400.00	370.21	287.55	405.94
$\Phi_9 = \{\{p_1, p_3\}, \{p_2, p_4\}\}$	301.18	233.68	30.51	248.90	379.91	352.07	278.00	359.94	416.71	368.07	278.00	375.94
$\Phi_{10} = \{\{p_1, p_4\}, \{p_2, p_3\}\}$	369.03	246.09	57.48	302.19	400.00	362.39	290.27	378.08	400.00	402.39	290.27	378.08
$\Phi_{11} = \{\{p_1, p_2, p_3\}, \{p_4\}\}$	267.02	211.93	-49.37	333.16	369.82	352.31	260.09	378.08	406.62	392.31	260.09	378.08
$\Phi_{12} = \{\{p_1, p_2, p_4\}, \{p_3\}\}$	328.12	192.77	129.33	203.15	385.68	337.75	370.21	345.62	416.08	358.55	370.21	358.42
$\Phi_{13} = \{\{p_1, p_3, p_4\}, \{p_2\}\}$	270.74	317.94	-41.42	230.26	369.90	370.21	257.45	357.54	406.70	370.21	257.45	395.94
$\Phi_{14} = \{\{p_2, p_3, p_4\}, \{p_1\}\}$	400.00	207.22	30.40	221.83	400.00	346.34	281.82	351.49	400.00	386.34	281.82	389.89
$\Phi_{15} = \{\{p_1, p_2, p_3, p_4\}\}$	246.28	182.75	-51.44	201.08	363.90	340.34	260.04	345.57	400.70	369.14	269.64	383.97

TABLE 5.4: Shapley value of the different approaches for the decisions in the simulation experiment (when the shippers own drones, i.e., the initial cost is set to S\$0))

Coalition Structure	CoDDD				CoSDD				BCoSDD			
	P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4
$\Phi_1 = \{\{p_1\}, \{p_2\}, \{p_3\}, \{p_4\}\}$	369.41	217.94	29.33	233.16	371.76	270.21	198.09	278.08	371.76	270.21	198.09	278.08
$\Phi_2 = \{\{p_1, p_2\}, \{p_3\}, \{p_4\}\}$	315.98	164.51	29.33	233.16	357.54	255.98	198.09	278.08	375.14	255.98	198.09	278.08
$\Phi_3 = \{\{p_1, p_3\}, \{p_2\}, \{p_4\}\}$	235.89	217.94	-104.20	233.16	317.63	270.21	143.95	278.08	352.83	270.21	143.95	278.08
$\Phi_4 = \{\{p_1, p_4\}, \{p_2\}, \{p_3\}\}$	315.77	217.94	29.33	179.52	350.72	270.21	198.09	257.03	371.52	270.21	198.09	257.03
$\Phi_5 = \{\{p_2, p_3\}, \{p_1\}, \{p_4\}\}$	369.41	146.09	-42.53	233.16	371.76	263.07	190.95	278.08	371.76	303.07	190.95	278.08
$\Phi_6 = \{\{p_2, p_4\}, \{p_1\}, \{p_3\}\}$	369.41	133.75	29.33	148.97	371.76	252.19	198.09	260.06	371.76	268.19	198.09	276.06
$\Phi_7 = \{\{p_3, p_4\}, \{p_1\}, \{p_2\}\}$	369.41	217.94	-43.14	160.70	371.76	270.21	187.55	267.54	371.76	270.21	187.55	305.94
$\Phi_8 = \{\{p_1, p_2\}, \{p_3, p_4\}\}$	315.98	164.51	-43.14	160.70	357.54	255.98	187.55	267.54	375.14	255.98	187.55	305.94
$\Phi_9 = \{\{p_1, p_3\}, \{p_2, p_4\}\}$	235.89	133.75	-104.20	148.97	317.63	252.19	143.95	260.06	352.83	268.19	143.95	276.06
$\Phi_{10} = \{\{p_1, p_4\}, \{p_2, p_3\}\}$	315.77	146.09	-42.53	179.52	350.72	263.07	190.95	257.03	371.52	303.07	190.95	257.03
$\Phi_{11} = \{\{p_1, p_2, p_3\}, \{p_4\}\}$	202.95	113.15	-155.55	233.16	302.37	247.81	135.78	278.08	339.17	287.81	135.78	278.08
$\Phi_{12} = \{\{p_1, p_2, p_4\}, \{p_3\}\}$	262.93	80.91	29.33	95.92	336.55	238.02	198.09	239.07	363.75	250.82	198.09	258.27
$\Phi_{13} = \{\{p_1, p_3, p_4\}, \{p_2\}\}$	197.91	217.94	-161.00	122.72	301.07	270.21	187.90	250.98	337.87	270.21	137.90	289.38
$\Phi_{14} = \{\{p_2, p_3, p_4\}, \{p_1\}\}$	369.41	108.03	-68.86	122.64	371.76	264.52	199.89	268.99	371.76	294.92	217.49	296.19
$\Phi_{15} = \{\{p_1, p_2, p_3, p_4\}\}$	172.45	82.57	-159.34	92.14	297.95	261.41	161.29	264.57	333.15	272.61	183.69	294.97

5.6.5 Belief update mechanism

The belief update mechanism is used in the BCoSDD framework in multiple iterations. Belief probability of the shippers are constantly being updated every time a delivery is done. The cooperative shipper can observe the behaviour of the other shippers. The BCoSDD framework is not a real-time system. I evaluate the belief update mechanism of the BCoSDD framework by assuming that shippers p_1 and p_2 always deliver all packages when they cooperate, i.e., good cooperative shippers, while shippers p_3 and p_4 will misbehave with a probability of 0.25, i.e., they are the bad cooperative shippers for 25% of the time. Therefore, the shipper whose packages are unsuccessfully delivered by shippers p_3 and p_4 needs to pay the penalty. All the shippers are initialised to believe that the other shippers will deliver all packages, i.e., $P(b_{pq} = T^G) = 1$ and $P(b_{pq} = T^B) = 0$.

Figure 5.11 shows the belief probabilities and the stable coalition structure over iterations, i.e., until the belief probabilities converge. From Figure 5.11b, at the first iteration, shippers p_1 and p_3 cooperate with each other, and shippers p_2 and p_4 cooperate with each other, i.e., Φ_9 . Since shipper p_1 transfers 11 packages to shipper p_3 and shipper p_3 only delivers 8 of them, the belief probability of shipper p_1 to shipper p_3 decreases, i.e., $P(b_{1,3} = T^G) = 0.92$, at the second iteration. Note that shipper p_3 does not transfer any package to shipper p_1 at the first iteration. Similarly, the belief probability of shipper p_2 to shipper p_4 also decreases, i.e., $P(b_{2,4} = T^G) = 0.92$ at the second iteration. Thus, at the second iteration, the stable coalition structure changes to Φ_6 . Shipper p_1 does not want to cooperate with shipper p_3 anymore, but shipper p_2 still cooperates with shipper p_4 . The stable coalition structure is also updated according to the belief probabilities. The belief probabilities are continuously updated based on the observation of the unsuccessful delivery. After the sixth iteration, the belief probabilities converge, and all the shippers have learned about the other shippers' behaviour. In this case, all the shippers do not want to cooperate with each other, i.e., coalition structure Φ_1 is reached.

FIGURE 5.11: Belief update mechanism, where shippers p_1 and p_2 always deliver packages, shippers p_3 and p_4 deliver only 75% of packages that belong to the other cooperative shippers.

5.7 Conclusion

The Bayesian Shipper Cooperation in Stochastic Drone Delivery (BCoSDD) framework has been proposed. The uncertainties of drone breakdown and shipper misbehaviour have been considered in this framework. The goals of this framework are to help multiple shippers (i) plan their package delivery by either using drones to serve customers or outsourcing the delivery to a carrier and (ii) decide whether the shippers should cooperate or not to minimise the delivery cost. To address the problem, I have proposed (i) package assignment, (ii) cost management, and (iii) shipper cooperation management in the BCoSDD framework. To achieve [Objective 2](#), I have formulated the package assignment as the multistage stochastic programming. In the shipper cooperation management, I have used the merge and split algorithm to reach a stable coalition structure, and thus [Objective 5](#) is achieved. For [Objective 6](#), the belief update mechanism has been proposed. The Bayesian game approach has been utilised to address the uncertainty of shippers' misbehaviour. For a shipper that does not have prior knowledge about the other cooperative shippers, the belief update mechanism has been proposed. The belief probabilities will be updated according to the observation when each delivery is performed. At the end, the extensive experiments of the BCoSDD framework have been conducted by using two datasets, i.e., Solomon Benchmark suite and a real dataset from a Singapore logistics industry. Based on the experiments, I can conclude that the BCoSDD framework can achieve the most effective solution, e.g.,

the lowest cost, compared with other baseline approaches.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

Different shippers may have different business goals and missions. To help diverse shippers with different assumptions, I address the objectives, which are presented in Section 1.2, by proposing three frameworks. The frameworks are listed as below. The goal is to help shippers schedule and plan their delivery by providing a set of potential solutions.

- Joint ground and aerial delivery service optimisation and planning (GADOP) framework.
- Multi-objective optimisation for drone delivery (MODD) framework
- Bayesian shipper cooperation in stochastic drone delivery (BCoSDD) framework

The aim of all the proposed frameworks is to help single/multiple shippers to (i) make the decision on the delivery mode that should be used and (ii) plan the delivery route. If a delivery drone breaks down, the shipper needs to pay the penalty for the unsuccessful delivery. In some cases, the penalty can raise the total cost, making it more expensive than using the ground-based delivery mode or the outsourcing mode. In addition, drones may not be able to reach some customers, e.g., due to flying distance limit. Therefore, I have considered the

joint of truck delivery and drone delivery in the GADOP framework, and the joint drone delivery and outsourcing mode in the MODD framework and the BCoSDD framework. Therefore, [Objective 1](#) has been met by presenting trade-offs between (i) truck delivery and drone delivery and (ii) drone delivery and outsourcing mode.

In all the proposed optimisation frameworks, the uncertainty of drone delivery has been taken into account by considering takeoff and breakdown conditions. The GADOP and MODD optimisation have been formulated as a three-stage stochastic integer programming model. After that, I have extended the three-stage model into the multi-stage stochastic optimisation in the package assignment component of the BCoSDD framework to dealing with the uncertainty in drone breakdown conditions. As a result, [Objective 2](#) has been accomplished.

To reduce the computational time in package delivery planning, which is [Objective 3](#), I have presented the decomposition algorithm, which is shown in Algorithm 1 and its formulations in the GADOP framework. The decomposition method is called L-shape method.

The GADOP framework and MODD framework are dedicated for a single shipper. I have presented single objective optimisation and multi-objective optimisation in the GADOP and the MODD, respectively. For the GADOP, the objective is to minimise the total delivery cost. For the MODD, the objectives are (i) to minimise the total delivery cost, (ii) to minimise the percentage of unsuccessful delivered packages, and (iii) to maximise the reward of on-time delivery. As a result, I have met [Objective 4](#) by introducing the methodology, which is summarised as the four steps in Section 4.2.4.

Additionally, the BCoSDD framework has proposed to help multiple shippers plan their package delivery and also help them decide whether they should cooperate with each other to minimise the delivery cost. The BCoSDD framework is an extension of the MODD framework, where the shippers can use drone delivery mode and outsourcing mode. Different from the MODD framework, I have formulated the BCoSDD as a single objective optimisation in order to concentrate on the cooperation. Algorithm 2 has been proposed as part of the BCoSDD framework to help the shippers decide how they should form the cooperation and thus [Objective 5](#) has been achieved.

In the BCoSDD framework, the uncertainty of shipper misbehaviours has been considered to help the shippers make a decision on with whom they should cooperate. For the shipper that does not have prior knowledge about the other cooperative shippers, I have proposed the belief update mechanism, which is presented in Algorithm 3, and thus Objective 6 has been met. The belief probabilities will be updated according to the observation when each delivery is performed.

The performance evaluation of both Solomon benchmark suite and Singapore real road network has been conducted. The simulation results have shown that the GADOP framework always achieves the lower total payment than that of the expected value formulation (EVF) and the Parallel Drone Scheduling Traveling Salesman Problem (PDSTSP). The L-shape decomposition method has been adopted to enhance the solvability of the optimisation problem. The experiments have shown that the decomposed version performed faster than the original version due to the fact that a smaller problem is less complicated and easier to be solved. For the MODD framework, I have analysed the Pareto frontiers. The set of solutions are listed for the shipper to select based on his/her preference. For the BCoSDD framework, the experiments show that shipper cooperation can significantly help the shippers to reduce the delivery cost and enhance the number of serving customers by expanding the coverage area of drones. I can conclude that the BCoSDD framework can achieve the most effective solution, e.g., the lowest cost, compared to other baseline approaches, i.e., static and deterministic approaches.

To sum up, the results of the proposed frameworks clearly show that (i) using drones in the package delivery can significantly reduce the total delivery cost and (ii) the frameworks with the uncertainties can achieve lower delivery cost than that of the deterministic planning system.

6.2 Future works

Since drone delivery is an emerging research topic, there are several future research directions. This section presents three possible future directions in drone delivery research.

Speedup computational times: Package/parcel delivery problem, i.e., VRP, is a non-polynomial hardness problem, which can be referred as a non-deterministic

polynomial acceptable problems or NP-hard problem in short. While the problem is complicated to solve in general, the size of the problems are usually large, e.g., hundreds or thousands of customers when considered in the real industries. Moreover, once multiple scenarios are considered to address randomness, the size of the problem increases significantly. Therefore, acceleration of computational time is one of the important challenges. This challenge can be addressed by multiple alternative techniques including decomposition and sampling average.

In the future, I can apply the sample-average approximation (SAA) approach [10] to relax the drone package delivery and obtain the solution faster. The SAA approach aims to relax a stochastic optimisation problem by using sets of scenarios, which are referred to as samples. In each sample, one or more scenarios are generated by Monte Carlo sampling technique. After that, a sample average is used to approximate the objective value of a stochastic optimisation problem, and thus the sample is used to solve deterministic optimisation problem. The globally optimal solution can be achieved if the number of scenarios is large enough.

In addition, one of the popular decomposition techniques has been adopted in the GADOP framework, which is to decomposes the problem to multiple sample sub-problems based on stages. I can extend this study by further reducing the computational time in a multi-thread processing, i.e., parallel processing, because the sub-problems are already divided and independent.

Parallel computing is one of the major approaches in high performance computing research. After successfully applying L-shape decomposition and SAA techniques to an optimisation problem, each sub-problem in decomposition and each sample in SAA can be independently solved. Therefore, there are opportunities to solve the problems in parallel, as multiple tasks are split up to be solved simultaneously by multiple processors (computer cores) in single/multiple computer nodes. Many programmers frequently implement a program that use only one processor core, but today computers have multiple cores. Thus, they do not fully utilise the power of computers. If I extend the problem to be a parallel programming, I can fully utilise the power of computers and may forward the program to be executed on a supercomputer. Currently, many large scale problems are solved by supercomputers (clusters), which have multiple nodes and thousands of processors.

Real-time stochastic drone delivery: According to the literature review based on the combination of stochastic and dynamic VRP, which has been presented in 2.4.1.3, I believe that this is an area that should be further investigated. Some researchers refer to dynamic systems as online or real-time systems. The survey [140] has shown that only a few number of existing studies considered both stochastic and dynamic parameter. Most studies in [140] considered only one uncertainty parameter (e.g. a study consider either traffic traveling time, customer demand, or package weight unit). Recently, I proposed a re-route stochastic package delivery planning dedicated for a ground-based delivery system [121]. In fact, this work can be extended to a multiple stage stochastic delivery based on the number of customers to improve the delivery plan. Moreover, to the best of our knowledge, none of the previous works on real-time stochastic delivery has experimented in the real system. In addition, none of the existing works is dedicated for drone delivery.

Cooperative drone delivery for multiple shippers: Shipper can cooperate to share resources, e.g., customers, drone, trucks, depots. They are likely to achieve more efficient delivery plan when they share resources than when they do not share them. I have proposed the BSoCDD framework to help shippers to cooperate. However, the BSoCDD framework considers only one objective, which is to minimise the delivery cost. Single objective is simpler to address compared to multiple objectives. In the future, the cooperative delivery can be extended to consider multiple objectives, which are similar to the objectives in the MODD framework, especially, when different shippers prefer a different major objective. When multiple objectives are considered, the difficulty of forming coalition structures and sharing pools increases.

In addition, shippers can share their customers by offering/bidding on the delivery fee, i.e., auction approach, to reveal minimum data as possible. A cooperative shipper can be either a common shipper or an occasional shipper. An occasional shipper is a shipper who is available to perform a delivery on a specific time slot and may or may not continue be a shipper in the future. Examples of an occasional shipper are Grab driver, Food Panda carrier, Lineman. According to the literature review in Chapter 2.5.2, the auction approach has been applied to VRP, but all of them were dedicated to the ground-based delivery mode. Moreover, the previous works have not

taken any uncertainty into account. Therefore, I can further explore shippers collaboration in a stochastic drone delivery with an auction approach.

Appendix A

Other Optimisation Techniques

A.1 Expected value formulation (EVF)

The EVF objective function is similar to the original objective function in (3.2), but only the formulation of the $\mathbb{E}(\mathcal{L}(\hat{X}_{i,d}))$ is different. The equation is indicated in (A.1), where $\mathbb{P}(d)$, $\mathbb{P}(i)$, and $\mathbb{M}(d)$ denote the probability that drone d cannot take off, the probability that breakdown occurs while serving customer i , and the probability that drone d needs to be repaired, respectively. The EVF subjects to the constraints in (3.5)-(3.23) and (3.28)-(3.37).

$$\begin{aligned}\mathbb{E}(\mathcal{L}(\hat{X}_{i,d})) &= \sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} \hat{c}_i^{(r)} \hat{X}_{i,d} \mathbb{P}(d) + \sum_{d \in \mathcal{D}} p \hat{X}_{i,d} (1 - \mathbb{P}(d)) \\ &\quad + \sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} p \hat{X}_{i,d} (1 - \mathbb{P}(d)) \mathbb{P}(i) \\ &\quad + \sum_{d \in \mathcal{D}} m (1 - \mathbb{P}(d)) \mathbb{M}(i)\end{aligned}\tag{A.1}$$

A.2 Parallel drone scheduling traveling salesman problem (PDSTSP)

The PDSTSP can handle multiple drones with only one truck. It does not take the serving time constraint and the constraints of traveling limits ,i.e., time and distance, into account. Furthermore, the the uncertainties i.e., takeoff and breakdown conditions, are ignored in the PDSTSP. We adapt the PDSTSP objective function, which is proposed in [11], from minimising the delivery time to minimising the total payment. The new objective function is presented in (A.2). The two delivery time constraints in [11] are replaced by the constraints in (A.3) and (A.4). The constraint in (A.3) ensures that the initial cost of the truck is paid when the truck is used. Similarly, the constraint in (A.4) ensures that the initial cost of drones are paid when the drones are used. The rest of the constraints in [11] are kept as the original.

Minimize:

$$\bar{c}^{(i)}\bar{W} + \sum_{d \in \mathcal{D}} \hat{c}_d^{(i)} \widehat{W}_d + \sum_{\substack{i', j' \in \\ \mathcal{C} \cup \{0\}}} \bar{c}_{i', j'}^{(r)} V_{i', j'} + \sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} \hat{c}_i^{(r)} \widehat{X}_{i, d} \quad (\text{A.2})$$

$$\sum_{i \in \mathcal{C}} V_i \leq \Delta \bar{W}, \quad \forall i \in \mathcal{C} \quad (\text{A.3})$$

$$\sum_{i \in \mathcal{C}} \widehat{X}_{i, d} \leq \Delta \widehat{W}_d, \quad \forall d \in \mathcal{D} \quad (\text{A.4})$$

List of Author's Publications

Journal Articles

- **S. Sawadsitang**, D. Niyato, P. S. Tan, and P. Wang, "Joint ground and aerial package delivery services: A stochastic optimization approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2241-2254, June 2019.
- **S. Sawadsitang**, D. Niyato, P. S. Tan, P. Wang, S. Nutanong, "Shipper Cooperation in Stochastic Drone Delivery: A Dynamic Bayesian Game Approach", Under reviewed in *IEEE Transactions on Vehicular Technology*. Available: <https://arxiv.org/abs/2002.03118>

Conference Proceedings

- **S. Sawadsitang**, D. Niyato, P. S. Tan, S. Nutanong, "Multi-objective Optimization in Drone Delivery," 2019 IEEE 90th Vehicular Technology Conference, Honolulu, Hawaii, USA.
- **S. Sawadsitang**, D. Niyato, K. Suankaewmanee, P. S. Tan, "Re-route package pickup and delivery planning with random demands," 2019 IEEE 90th Vehicular Technology Conference, Honolulu, Hawaii, USA.
- N. Ngoenriang, **S. Sawadsitang**, C. Leangsuksun, D. Niyato, and P. S. Tan, "Joint vehicle routing and loading in delivery planning: A stochastic programming approach," 2019 IEEE 88th Vehicular Technology Conference, Kuala Lumpur, Malaysia.

- **S. Sawadsitang**, D. Niyato, P. S. Tan, and P. Wang, “Supplier cooperation in drone delivery,” 2018 87th IEEE Vehicular Technology Conference, Chicago, USA.
- K. Suankaewmanee, D.T. Hoang, D. Niyato, **S. Sawadsitang**, P. Wang, and Z. Han “Performance analysis and application of mobile blockchain”, International Conference on Computing, Networking and Communication (ICNC), Hawaii, USA.
- **S. Sawadsitang**, S. Jiang, D. Niyato, and P. Wang “Optimal Stochastic Package Delivery Planning with Deadline: A Cardinality Minimization in Routing”, 2017 IEEE 86th Vehicular Technology Conference, Toronto, Canada, pp. 1-7.
- **S. Sawadsitang**, R. Kaewpuang, S. Jiang, D. Niyato, and P. Wang, “Optimal Stochastic Delivery Planning in Full-Truckload and Less-Than-Truckload Delivery”, 2017 IEEE 85th Vehicular Technology Conference, Sydney, Australia, pp. 1-5.

Bibliography

- [1] A. El Fallahi, C. Prins, and R. Wolfler Calvo. “A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem”. *Computers & Operations Research*, Vol. 35 (5), pp. 1725-1741, 2008. 21
- [2] A. Kimms and I.Kozeletskyi, “Core-based cost allocation in the cooperative traveling salesman problem,” *European Journal of Operational Research*, vol. 248, pp. 910-916, 2016. 34, 43
- [3] A. Rizzoli, R. Montemanni, E. Lucibello, L. Gambardella, “Ant colony optimization for real-world vehicle routing problems”. *Swarm Intelligence*, Vol. 1, pp. 135-151, 2007. 30
- [4] A. Robinson, “What Is Last Mile Logistics & Why Are More Shippers Looking at This Transportation Function?” [Online]. Available: <https://cerasis.com/last-mile-logistics/> 2
- [5] B. De Backer , V. Furnon, P. Prosser, P. Kilby, and P. Shaw. “Local search in constraint programming: Application to the vehicle routing problem”. Proc.CP-97 Workshop Indust.Constraint-Directed Scheduling, Schloss Hagenberg, Austria, pp. 1-15, 1997. 1
- [6] B. Eksioglu, A. V. Vural b, and A. Reisman. “The vehicle routing problem: A taxonomic review”. *Computers & Industrial Engineering*, Vol. 57, pp. 1472-1483, 2009. 13
- [7] B. L. Golden, S. Raghavan, and E. Wasil. “The vehicle routing problem: latest advances and new challenges”. *Springer*, 2008. 13
- [8] B. N. Coelho, V. N. Coelho, I. M. Coelho, L. S. Ochi, R. Haghazadeh K., D. Zuidema, M. S.F. Lima, A. R. Costa, “A multi-objective green UAV routing problem,” *Computers & Operations Research*, Vol. 88, 2017, PP. 306-315. 32, 33, 42
- [9] Business Insider. [Online]. Available:<https://www.businessinsider.com/commercial-uav-market-analysis-2017-8/?IR=T> 2
- [10] B. Verweij S. Ahmed, A. J. Kleywegt, G. Nemhauser, and A. Shapiro. “The Sample Average Approximation Method Applied to Stochastic Routing Problems: A Computational Study”. *Computational Optimization and Applications*, Vol. 24, Issue 2–3, pp. 289–333, 2003. 142

- [11] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 86-109, May 2015. [5](#), [25](#), [26](#), [41](#), [49](#), [76](#), [146](#)
- [12] C. Groer, B. Golden and E. Wasil, "A parallel algorithm for the vehicle routing problem". *INFORMS Journal on Computing*, Vol. 23 (2), pp. 315-330, 2011. [39](#)
- [13] C. J. L. Arachchige, S. Venkatesan, and N. Mittal, "An asynchronous neighbor discovery algorithm for cognitive radio networks," in *Proc. IEEE Symp. on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, Chicago, IL, USA, Oct. 2008. [118](#)
- [14] C. Prins, "A simple and effective evolutionary algorithm for the vehicle routing problem". *Computers & Operations Research*, Vol. 31 (12), pp. 1985-2002, 2004. [39](#)
- [15] C. Prins, "A GRASP – evolutionary local search hybrid for the vehicle routing problem". In: F. Pereira and J. Tavares, (Eds.), "Bio-Inspired Algorithms for the Vehicle Routing Problem". Springer, Berlin, Heidelberg, pp. 35-53, 2009. [39](#)
- [16] C. Sabo, D. Kingston and K. Cohen, "A Formulation and Heuristic Approach to Task Allocation and Routing of UAVs under Limited Communication," *Unmanned Systems*, vol. 2, pp. 1-17, Dec 2013. [24](#)
- [17] C. Tarantilis, "Solving the vehicle routing problem with adaptive memory programming methodology". *Computers & Operations Research*, Vol. 32 (9), pp. 2309-2327, 2005. [39](#)
- [18] C. Watson-Gandy, and P. Dohm. "Depot location with van salesmen – a practical approach". *Omega*, Vol. 1, pp. 321-329, 1973. [20](#)
- [19] C.-W. Chu, "A heuristic algorithm for the truckload and less-than-truckload problem", *European Journal of Operational Research*, vol. 165 (3), pp. 657-667, 2005. [23](#), [41](#)
- [20] C. Lin, K. L. Choy, G. T. S. Ho, S. H. Chung and H. Y. Lam, "Survey of Green Vehicle Routing Problem: Past and future trends," *Expert Systems with Applications*, vol. 41, pp. 1118-1138, 2014. [xvii](#), [13](#), [17](#), [18](#), [19](#), [20](#), [21](#), [22](#), [29](#), [39](#)
- [21] C. Triki, S. Oprea, P. Beraldi, T. G. Crainic, "The stochastic bid generation problem in combinatorial transportation auctions". *European Journal of Operational Research*, Vol. 236, pp. 991-999, 2014. [35](#)
- [22] D. Chattopadhyay, "Application of General Algebraic Modeling System to Power System Optimization," *IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 15-22, Feb 1999. [65](#), [91](#), [125](#)

- [23] D.J. Higham, N.J. Higha, “MATLAB guide,” Vol. 150. Siam; 2016. [129](#)
- [24] D. Mester and O. Bräysy, “Active-guided evolution strategies for large-scale capacitated vehicle routing problems”. *Computers & Operations Research*, Vol. 34 (10), pp. 2964-2975, 2007. [39](#), [40](#)
- [25] D. Pisinger, S. Ropke, “A general heuristic for vehicle routing problems”. *Computers & Operations Research*, Vol. 34 (8), pp. 2403-2435, 2007. [39](#)
- [26] E. Chajakis, and M. Guignard, “Scheduling deliveries in vehicles with multiple compartments”. *Journal of Global Optimization*, Vol. 26, pp. 43-78, 2003. [21](#)
- [27] E. Taillard, P. Badeau, M. Gendreau, F. Guertin, J. Potvin, “A tabu search heuristic for the vehicle routing problem with soft time windows”. *Transportation Science*, Vol. 31 (2), pp. 170-186, 1997. [30](#)
- [28] E. Zachariadis and C. Kiranoudis, “A strategy for reducing the computational complexity of local search-based methods for the vehicle routing problem”. *Computers & Operations Research*, Vol. 37 (12), pp. 2089-2105, 2010. [39](#)
- [29] F. A. Tillman. “The multiple terminal delivery problem with probabilistic demands”. *Transportation Science*, Vol. 3, pp. 192-204, 1969. [20](#)
- [30] F. Guerriero, R. Surace, V. Loscr, E. Natalizio, “A multi-objective approach for unmanned aerial vehicle routing problem with soft time windows constraints,” *Applied Mathematical Modelling*, Vol. 38, Issue 3, 2014, PP. 839-852. [32](#), [33](#), [42](#), [89](#)
- [31] F. Li, B. Golden, E. Wasil, “Very large-scale vehicle routing: new test problems, algorithms, and results”. *Computers & Operations Research*, Vol. 32 (5), pp. 1165-1179, 2005. [39](#)
- [32] F. Schenkelberg, “How reliable does a delivery drone have to be?,” *2016 Annual Reliability and Maintainability Symposium (RAMS)*, Tucson, Arizona, USA, 2016, pp. 1-5. [4](#), [49](#)
- [33] G. B. Dantzig, and D. R. Fulkerson, “Minimizing the number of tankers to meet a fixed schedule”. *Naval Research Logistics Quarterly*, pp. 217-222, 1954. [22](#)
- [34] G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia and G. Laporte, “Static Pickup and Delivery Problems: A Classification Scheme and Survey,” *TOP*, vol. 15, pp. 1-31, Jul 2007. [13](#), [20](#)
- [35] G. Dantzig, J. Ramser. “The truck dispatching problem”. *Management Science*, Vol. 6, pp. 80-91, 1959. [13](#)
- [36] G. Kuyzu, C. G. Akyol, O. Ergun and M. Savelsbergh, “Bid price optimization for truckload carriers in simultaneous transportation procurement auctions”. *Transportation Research Part B: Methodological*, Vol. 73, pp. 34-58, 2015. [35](#)

- [37] G. Laporte, and F.V. Louveaux. "The integer Lshaped method for stochastic integer programs with complete recourse". *Operations Research Letters* Vol. 13, pp. 133-142, 1993. [28](#)
- [38] G. Laporte, M. Gendreau, J. Y. potvin, and F. Semet, "Classical and modern heuristics for the vehicle routing problem," *International transactions in Operational Research*, vol. 7, pp. 285-300, 2000. [13](#), [38](#)
- [39] G. Laporte, "The Vehicle Routing Problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, pp. 345-358, 1992. [13](#), [36](#), [37](#), [38](#)
- [40] G. Laporte and Y. Nobert, "Exact algorithms for the vehicle routing problem". North-Holland Mathematics Studies, Vol. 312, pp. 147 - 184, 1987. [36](#), [37](#), [38](#)
- [41] "Global last mile delivery market: global size, trends, competitive, historical & forecast analysis, 2019-2025". *Brand essence*, 2019. [1](#)
- [42] G. Mavrotas, "Effective implementation of the ϵ -constraint method in Multi-Objective Mathematical Programming problems," *Applied Mathematics and Computation*, Vol. 213, Issue 2, 2009, PP. 455-465. [90](#)
- [43] G. Nagy, and S. Salhi. "Location-routing: issues, models and methods". *European Journal of Operational Research*, Vol. 177, pp. 649-672, 2007. [20](#)
- [44] Google. [Online]. Available: <https://developers.google.com/maps/documentation/distance-matrix/> [66](#)
- [45] G. Perboli, R. Tadei, and D. Vigo. "The two-echelon capacitated vehicle routing problem: models and math-based heuristics". *Transportation Science*, Vol. 45, pp. 364-380, 2011. [21](#)
- [46] G. Zhang, K. Smilowitz, and A. Erera. "Dynamic planning for urban drayage operations." *Transportation Research Part E: Logistics and Transportation Review*, vol. 47, no. 5, pp. 764-777, 2011. [31](#)
- [47] H. B. Bendall and A. F. Stent, "A scheduling model for a high speed container-ship service: A hub and spoke short-sea application". *International journal of maritime economics*, Vol. 3, pp. 262-277, 2001. [22](#)
- [48] H. Lee, M. Boile, S. Theofanis, and S. Choo, "Game theoretical models of the cooperative carrier behavior," *KSCE Journal of Civil Engineering*, vol. 18, no. 5, pp 1528-1538, June 2014. [97](#)
- [49] H. GIRAY RESAT, "Design and Analysis of Novel Hybrid Multi-Objective Optimization Approach for Data-Driven Sustainable Delivery Systems," *IEEE Access*, Vol. 8, 2020, PP. 90280 - 90293. [32](#), [33](#)
- [50] H. Min, V. Jayaraman, and R. Srivastava. "Combined location-routing problems: a synthesis and future research directions". *European Journal of Operational Research*, Vol. 108, pp. 1-15, 1998. [20](#)

- [51] H. Psaraftis, “A dynamic-programming solution to the single vehicle many-to-many immediate request dial-a-ride problem”. *Transportation Science*, Vol. 14 (2) pp. 130-154, 1980. [29](#)
- [52] H. Wilson and H. Weissberg. “Advanced dial-a-ride algorithms research project:final report”. Technical Report. R76-20, Department of Civil Engineering, MIT, Cambridge, MA, 1976. [19](#)
- [53] I. F. A. Vis and R. Koster, “Transshipment of containers at a container terminal: An overview”. *European Journal of Operational Research*, Vol. 147, Issue 1, pp. 1-16, 2003. [23](#)
- [54] I. Hong, M. Kuby, A. T. Murray, “A range-restricted recharging station coverage model for drone delivery service planning,” *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 198-212, 2018. [26](#)
- [55] J. Darzentas and T. Spyrou, “Ferry traffic in the Aegean Islands: A simulation study”. *Journal of the Operational Research Society*, Vol. 47, pp. 203-216, 1996. [22](#), [23](#)
- [56] J. E. Mendoza, B. Castanier, C. Guret, A. L. Medaglia, A. L., and N. Velasco, “A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands”. *Computers & Operations Research*. Vol. 37, pp. 1886-1898, 2010. [28](#)
- [57] J. Highsmith and A. Cockburn, “Agile software development: The business of innovation.” *Computer*, Vol. 34, No. 9, pp. 120-127, 2001. [1](#)
- [58] J. Euch and H. Chabchoub , “Iterated Density Estimation with 2-opt local search for the vehicle routing problem with private fleet and common carrier”. *International Conference on Computers & Industrial Engineering (CIE)*, pp.1058-1063, 6-9 July 2009. [23](#), [41](#)
- [59] J.-F. Cordeau, M. Maischberger, “ A parallel iterated tabu search heuristic for vehicle routing problems”. *Computers & Operations Research*, Vol. 39 (9), pp. 2033-2050, 2012. [39](#)
- [60] J. Kratica, T. Kostic, D. Tomic, D. Dugosija, and V. Filipovic, “A Genetic Algorithm for the Routing and Carrier Selection Problem”, *Computer Science and Information Systems*, vol. 9, no. 1, pp. 49-62, 2012. [23](#), [41](#)
- [61] J. Jin, T. Crainic and A. Lokketangen, “A guided cooperative parallel tabu search for the capacitated vehicle routing problem”. In: *Proceedings of NIK’11*. pp. 49-60, 2011. [39](#)
- [62] J. Jin, T. Crainic and A. Lokketangen, “A parallel multi-neighborhood cooperative tabu search for capacitated vehicle routing problem”. *European Journal of Operational Research*, Vol. 222 (3), pp. 441-451, 2012. [39](#)

- [63] J. Park, S. Kim, and K. Suh, "A comparative analysis of the environmental benefits of drone-based delivery services in urban and rural areas." *Sustainability*, Vol. 10 No.3, pp 888, 2018. [2](#)
- [64] J. R. Birge and F. Louveaux, "Introduction to Stochastic Programming". New York: Springer, 2011. [41](#), [42](#), [58](#), [59](#), [62](#)
- [65] J. Renaud, G. Laporte, , and F. F. Boctor, "A tabu search heuristic for the multidepot vehicle routing problem". *Computers & Operations Research*, Vol. 23, pp. 229-235, 1996. [20](#)
- [66] J.S. Milton and J. Arnold, "Introduction to Probability and Statistics: Principles and Applications for Engineering and the Computing Sciences". *McGraw Hill*, 1995. [120](#)
- [67] J. Yang, P. Jaillet, H. Mahmassani, "Real-time multivehicle truckload pickup and delivery problems". *Transportation Science*, Vol. 38, (2), pp. 135-148, 2004. [29](#)
- [68] K. Akkarajitsakul, E. Hossain and D. Niyato, "Coalition-Based Cooperative Packet Delivery under Uncertainty: A Dynamic Bayesian Coalitional Game," *IEEE Transaction on Mobile Computing*, vol. 12, no. 2, pp. 371-385, 2013. [120](#)
- [69] K. Fagerholt, "Optimal fleet design in a ship routing problem". *International Transactions in Operational Research*, pp. 453-464, 1999. [22](#)
- [70] K. Fagerholt and B. Rygh, "Design of a sea-borne system for fresh water transport—A simulation analysis". *Belgian journal of operations research, statistics and computer science*, Vol. 40(3-4), pp. 137-146, 2002. [22](#), [23](#)
- [71] K. Fagerholt and H. Lindstad, "Optimal policies for maintaining a supply service in the Norwegian Sea". *OMEGA*, Vol. 28, pp. 269-275, 2000. [22](#)
- [72] K. Dorling, J. Heinrichs, G. G. Messier and S. Magierowski, "Vehicle Routing Problems for Drone Delivery," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 70-85, Jan 2017. [24](#), [26](#), [32](#)
- [73] K. Sundar and S. Rathinam, "Algorithms for Routing an Unmanned Aerial Vehicle in the Presence of Refueling Depots," in *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 1, pp. 287-294, 2014. [24](#), [25](#)
- [74] L. A. Haidari, S. T. Brown, M. Ferguson, E. Bancroft, M. Spiker, A. Wilcox, R. Ambikapathi, V. Sampath, D. L. Connor, B. Y. Lee, "The economic and operational value of using drones to transport vaccines," In *Vaccine*, vol. 34, pp. 4062-4067, Jul 2016. [49](#)
- [75] L. Evers, A. I. Barros, H. Monsuur and A. Wagelmans, "Online stochastic UAV mission planning with time windows and time-sensitive targets," *European Journal of Operational Research*, vol. 238, pp. 348-362, Oct 2014. [28](#)

- [76] L. Evers, T. Dollevoet and A. I. Barros, “Robust UAV mission planning,” *Annals of Operations Research*, vol. 222, pp. 293-315, Nov 2014. [28](#)
- [77] L. Gambardella, A. Rizzoli, F. Oliverio, N. Casagrande, A. Donati, R. Montemanni, E. Lucibello, “Ant colony optimization for vehicle routing in advanced logistics systems”. International Workshop on Modelling and Applied Simulation (MAS 2003), pp. 3-9, 2003. [30](#)
- [78] L. M. Hvattum, A. Løkketangen, and G. Laporte. “Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic.” *Transportation Science*, vol. 40, no. 4, pp. 421-438, 2006. [31](#)
- [79] Land Transportation Authority of Singapore, “Singapore Land Transportation Statistic in brief,” 2009. [Online]. Available: https://www.lta.gov.sg/content/dam/ltaweb/corp/PublicationsResearch/files/FactsandFigures/Statistics_in_Brief_2009.pdf [65](#)
- [80] M. A. Krajewska, H. Kopfer, G. Laporte, S. Ropke and G. Zaccour, “Horizontal cooperation among freight carriers: request allocation and profit sharing,” *Journal of the Operational Research Society*, vol. 59, pp. 1483-1491, 2008. [34](#), [43](#)
- [81] M.-C. Bolduc, J. Renaud, and F. Boctor, “A heuristic for the routing and carrier selection problem”, *European Journal of Operational Research*, vol. 183, no. 2, pp. 926-932, 1 December 2007. [23](#), [41](#)
- [82] M. Christiansen, K. Fagerholt and D. Ronen, “Ship routing and scheduling: status and perspectives”. *Transport Science*, Vol. 38, No. 1, pp. 1-8, 2004. [22](#), [23](#)
- [83] M. Crary, L. K. Nozick and L. R. Whitaker, “Sizing the U.S. destroyer fleet”. *European Journal of Operational Research*, Vol. 136, pp. 680–695, 2002. [22](#)
- [84] M. Dror, G. Laporte, and F. V. Louveaux. “Vehicle routing with stochastic demands and restricted failures”. *ZOR-Methods and Models of Operations Research*, Vol. 37, pp. 273-283, 1993. [28](#)
- [85] M. Frisk, M. Gothe-Lundgren, K. Jornsten and M. Ronnqvist, “Cost allocation in collaborative forest transportation,” *European Journal of Operational Research*, vol. 205, pp. 448-458, 2010. [34](#), [43](#)
- [86] M. Gendreau, F. Guertin, J.-Y. Potvin, E. Taillard, “Parallel tabu search for realtime vehicle routing and dispatching”. *Transportation Science* 33 (4), pp. 381-390, 1999. [30](#)
- [87] M. Gendreau, G. Laporte, and R. Seguin. “Stochastic vehicle routing”. *European Journal of Operational Research*, vol. 88, pp. 3-12, 1996. [27](#)
- [88] M. Gendreau, G. Laporte, and R. Seguin. “An exact algorithm for the vehicle routing problem with stochastic customers and demands”. *Transportation Science*, Vol. 29, pp. 143-155, 1995. [28](#)

- [89] M. Gendreau, G. Laporte and R. Seguin, “Stochastic vehicle routing,” *European Journal of Operational Research*, vol. 88, pp. 3-12, 1996. [13](#), [28](#), [40](#)
- [90] M. Guajardo and M. Ronnqvist, “A review on cost allocation methods in collaborative transportation,” *Intelligent Transportation in Operation Research*, vol. 23, pp. 371–392, 2016. [34](#), [43](#)
- [91] M. Gansterer and R. F. Hartl, “Request evaluation strategies for carriers in auction-based collaborations”. *OR Spectrum*, Vol. 38, pp. 3-23, 2016. [35](#), [36](#)
- [92] M. Mes, M. V. D. Heijden and P Schuur, “Interaction between intelligent agent strategies for real-time transportation planning”. *Central European Journal of Operations Research*, Vol. 21, pp. 337-358, 2013. [35](#)
- [93] M. M. Solomon, “Algorithms for the vehicle routing and scheduling problems with time window constraints,” *Operations Research*, vol. 35, no. 2, pp. 254-265, 1987. [30](#), [66](#), [125](#)
- [94] M. M. Solomon. “Time Window Constrained Routing and Scheduling Problem”. *Transportation Science*, Vol. 22, pp. 1-13, 1988. [13](#), [19](#)
- [95] M. O. Ball, A. Golden, A. Assad, and L. D. Bodin, “Planning for truck fleet size in the presence of a common-carrier option”. *Decision Sciences*, Vol. 14, pp. 103-120, 1983. [5](#), [23](#)
- [96] M. Pavone, N. Bisnik, E. Frazzoli, and V. Isler. “A Stochastic and Dynamic Vehicle Routing Problem with Time Windows and Customer Impatience”. *Mobile Network Application* Vol. 14, pp. 350-364, 2009. [28](#)
- [97] M. Pourahmadi, “Foundations of Time Series Analysis and Prediction Theory”. *John Wiley & Sons*, 2001. [122](#)
- [98] M. Reimann, K. Doerner and R. Hartl, “D-ants: savings based ants divide and conquer the vehicle routing problem”. *Computers & Operations Research*, Vol. 31 (4), pp. 563-591, 2004. [39](#)
- [99] M. Wen, J. Larsen, J. Clausen, J. F. Cordeau, and G. Laporte. “Vehicle routing with cross-docking. *Journal of the Operational Research Society*, Vol. 60, pp. 1708-1718, 2009. [21](#)
- [100] N. Christofides, A. Mingozi and P. Toth, “Space State Relaxation Procedures for the Computation of Bounds to Routing Problems”, *Networks* , Vol. 11, pp. 145 - 164, 1981. [37](#)
- [101] N. Christofides and S. Eilon, “An algorithm for the vehicle-dispatching problem”. *Operational Research Society*, Vol 2, No. 3, pp. 309-138. 1969 [36](#), [37](#)
- [102] N. Jozefowicz, F. Semet, and E.-G. Talbi, “Multi-objective vehicle routing problems,” *European Journal of Operational Research*, Vol. 189, Issue 2, 2008, PP. 293-309. [32](#), [42](#)

- [103] O. Braysy, and M. Gendreau. "Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms". *Transportation Science*, Vol. 39, pp. 104-118, 2005. 13, 19
- [104] O. Braysy, and M. Gendreau. "Vehicle Routing Problem with Time Windows, Part II: Metaheuristics". *Transportation Science*, Vol. 39, pp. 119-139, 2005. 13, 19
- [105] Advanced Routing and Scheduling Software - Paragon routing [Online]. Available: <https://www.paragonrouting.com/> 1
- [106] P. Liu, Y. Wu and N. Xu, "Allocating collaborative profit in less-than-truckload carrier alliance," *Journal of Service Science and Management*, vol. 3, pp. 143-149, 2010. 34, 43
- [107] P. Wu and D. A. Campbell, and T. Merz, "On-board multi-objective mission planning for unmanned aerial vehicles." *IEEE Aerospace Conference*, 7-14 March 2009, Big Sky, Montana. 32, 33, 42
- [108] R. Baldacci, E. Bartolini, and G. Laporte. "Some application of the generalized vehicle routing problem". *Journal of the Operational Research Society*, Vol. 61, pp. 1072-1077, 2010. 20
- [109] R. B. Myerson, "Game Theory, Analysis of Conflict," *Cambridge, MA: Harvard University Press*, 1991. 112, 114
- [110] R. C. Larson, "Transporting sludge to the 106-mile site: An inventory/routing model for fleet sizing and logistics system design". *Transportation Science*, Vol. 22(3), pp. 186-198, 1988. 22, 23
- [111] Rentalorrrysingapore.com [Online]. Available: <https://rentalorrrysingapore.com/lorry-dimensions/> 65
- [112] R. Kaewpuang, D. Niyato, P. S. Tan, and P. Wang, "Cooperative Management in Full-Truckload and Less-Than-Truckload Vehicle System," *IEEE Transaction on Vehicular Technology*, vol. 66, pp. 5707-5722, 2017. 34, 43
- [113] R. Montemanni, L.M. Gambardella, A.E. Rizzoli, A.V. Donati, "Ant colony system for a dynamic vehicle routing problem". *Journal of Combinatorial Optimization*, Vol. 10 (4), pp. 327-343, 2005. 30
- [114] R. Radharamanan and L. I. Choi, "A branch and bound algorithm for the travelling salesman and the transportation routing problems". *Computers & Industrial Engineering*, Vol. 11, Issues 1-4, PP. 236-240, 1986. 36, 37
- [115] S. Berger and C. Bierwirth, "Solutions to the request reassignment problem in collaborative carrier networks". *Transportation Research Part E: Logistics and Transportation Review*, Vol. 46, pp. 627-638, 2010. 35, 36
- [116] S.-C. Cho and A. N. Perakis, "Optimal liner fleet routing strategies". *Maritime Policy and Management*, Vol. 23(3), pp. 249-259, 1996 22, 23

- [117] S. Eilon, C.D.T. Watson-Candy and N. Christofides, "Distribution Management: Mathematical Modeling and Practical Analysis", *Griffin*, 1971. 37
- [118] S. Jung and H. Kim, "Analysis of Amazon Prime Air UAV delivery service". *Journal of Knowledge Information Technology and Systems*, Vol. 12, No. 2, pp. 253-266, 2017. 2
- [119] S. M. Ferrandez, T. Harbison, T. Weber, R. Sturges, R. Rich, "Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm," in *Journal of Industrial Engineering and Management*, Vol. 9, pp. 374-388, 2016. 25, 26
- [120] S. N. Kumar, and R. Panneerselvam. "A Survey on the Vehicle Routing Problem and Its Variants". *Intelligent Information Management*, pp. 66-74, 2012. 13
- [121] S. Sawadsitang, D. Niyato, K. Suankaewmanee, and P.S. Tan, "Re-route Package Pickup and Delivery Planning with Random Demands". *IEEE 90th Vehicular Technology Conference*, HI, USA, 2019. 31, 143
- [122] S. Sawadsitang, D. Niyato, P.S. Tan, and P. Wang, "Supplier Cooperation in Drone Delivery", *2018 IEEE 88th Vehicular Technology Conference (VTC fall)*, Chicago, USA, 2018. 5, 32, 87, 97
- [123] S. Sawadsitang, D. Niyato, P. Tan, and P. Wang, "Joint Ground and Aerial Package Delivery Services: A Stochastic Optimization Approach," in *IEEE Transactions on Intelligent Transportation Systems*. pp. 1-14, 2018. xiv, 32, 45
- [124] S. Sawadsitang, D. Niyato, P.S. Tan, P. Wang, and S. Nutanong, "Multi-Objective Optimization for Drone Delivery." *2019 IEEE 90th Vehicular Technology Conference (VTC Fall)*, HI, USA, 2019. xiv, 81
- [125] S. Sawadsitang, D. Niyato, P.S. Tan, P. Wang, and S. Nutanong, "Shipper Cooperation in Stochastic Drone Delivery: A Dynamic Bayesian Game Approach". (Under review). xv, 97
- [126] S. Sawadsitang, S. Jiang, D. Niyato, P. Wang, "Optimal Stochastic Package Delivery Planning with Deadline: A Cardinality Minimization in Routing," *2017 IEEE 86th Vehicular Technology Conference (VTC fall)*, Toronto, Canada, 2017. 78
- [127] S. Sawadsitang, R. Kaewpuang, S. Jiang, D. Niyato, P. Wang, "Optimal Stochastic Delivery Planning in Full-Truckload and Less-Than-Truckload Delivery," *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, Sydney, Australia, Jun 2017. 56, 66
- [128] Singpost. [Online]. Available: <http://www.singpost.com/> 66, 123
- [129] S. Zibaei, A. Hafezalkotob, and S. S. Ghashami. "Cooperative vehicle routing problem: an opportunity for cost saving". *Journal of Industrial Engineering International*, Vol. 12, Issue 3, pp. 271-286, 2016. 43

- [130] T. G. Crainic, and G. Laporte. “Planning models for freight transportation”. *European Journal of Operational Research*, Vol. 97, pp. 409–438, 1997.
- [131] T. G. Crainic, N. Ricciardi, and G. Storchi. “Models for evaluating and planning city logistics systems”. *Transportation Science*, Vol. 43, pp. 432–454, 2009. [21](#)
- [132] T. Jones, “International commercial drone regulation and drone delivery services”. No. RR-1718/3-RC. RAND, 2017. [2](#)
- [133] T. Huth, and D. C. Mattfeld. “Myopic and Anticipated Planning in Stochastic Swap Container Management.” *International Journal of Operations Research*, vol. 8, no. 1, pp. 3–22, 2011. [31](#)
- [134] Transparency Market Research , “E-commerce Logistics Market (Service Type - Warehousing (Mega Centers, Hubs/Delivery Centers, Returns Processing, Centers), Transportation (Air/Express Delivery, Freight/Rail, Trucking/Over Road, Maritime)) - Global Industry Analysis, Size, Share, Growth, Trends and Forecast 2016 - 2024,” July 2016. [1](#)
- [135] T. Vidal, T. Crainic, M. Gendreau, N. Lahrichi and W. Rei, “A hybrid genetic algorithm for multi-depot and periodic vehicle routing problems”. *Operations Research*, Vol. 60 (3), pp. 611–624, 2012. [39](#)
- [136] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins. “Heuristics for multi-attribute vehicle routing problems: A survey and synthesis”. *European Journal of Operational Research*, Vol. 231, pp. 1–21, 2013. [xvii](#), [13](#), [18](#), [21](#), [22](#), [38](#), [39](#)
- [137] U. Derigs, J. Gottlieb, J. Kalkoff, M. Piesche, F. Rothlauf, and U. Vogel. “Vehicle routing with compartments: applications, modeling and heuristics”. *Operational Research Spectrum*, Vol. 33, pp 885–914, 2011. [21](#)
- [138] U. Derigs, R. Kaiser, “Applying the attribute based hill climber heuristic to the vehicle routing problem”. *European Journal of Operational Research*, Vol. 177 (2), pp. 719–732, 2007. [39](#)
- [139] Unmanned Airspace, “Drone delivery services growing at extraordinary rate – new Unmanned Airspace survey,” 2019. [Online]. Available: <https://www.unmannedairspace.info/uncategorized/drone-delivery-services-growing-at-extraordinary-rate-new-unmanned-airspace-survey> [2](#)
- [140] U. Ritzinger, J. Puchinger, and R. F. Hartl. “A survey on dynamic and stochastic vehicle routing problems.” *International Journal of Production Research*, vol. 54, no. 1, pp. 215–231, 2016 [27](#), [31](#), [143](#)
- [141] V. Pillac, C. Gueret, and A. L. Medaglia. “An event-driven optimization framework for dynamic vehicle routing.” *Decision Support Systems*, vol. 54, no. 1, pp. 414–423, 2012. [27](#), [31](#)

- [142] V. Pillac, M. Gendreau, C. Gueret and A. L. Medaglia. “A review of dynamic vehicle routing problems,” *European Journal of Operational Research*, vol. 225, pp. 1-11, 2013. [13](#), [29](#)
- [143] V. Pillac, M. Gendreau, C. Gueret, and A. L. Medaglia. “A review of dynamic vehicle routing problems”. *European Journal of Operational Research*, vol. 225, pp. 1-11, 2013. [27](#)
- [144] W. B. Powell, “Approximate dynamic programming: solving the curses of dimensionality”. Wiley Series in Probability and Statistics, vol. 703, WileyInterscience, Hoboken, New Jersey, 2007. [29](#)
- [145] W. Saad, Z. Han, R. Zheng, A. Hjørungnes, T. Basar and H. V. Poor, “Coalitional Games in Partition Form for Joint Spectrum Sensing and Access in Cognitive Radio Networks,” in *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 2, pp. 195-209, April 2012. [116](#), [117](#), [118](#)
- [146] W. Yoo, E. Yu, and J. Jung. “Drone delivery: Factors affecting the public’s attitude and intention to adopt.” *Telematics and Informatics*, Vol. 35, No. 6, pp. 1687-1700, 2018. [2](#)
- [147] X. Lia, P. Tiana, and S. C.H.Leung. “Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm”. *International Journal of Production Economics*, Vol. 125, pp. 137-145, 2010. [28](#)
- [148] X. Wang, S. Poikonen, and B. Golden. ”The vehicle routing problem with drones: Several worst-case results.” *Optimization Letters*, col. 11.4, pp. 679-697, 2017. [26](#)
- [149] Y. Han, J. Li, Z. Liu, C. Liu and J. Tian, “Metaheuristic algorithm for solving the multi-objective vehicle routing problem with time window and drones,” *International Journal of Advanced Robotic Systems*, 2020, PP.1-14. [32](#), [33](#)
- [150] Y. Liu, “An optimization-driven dynamic vehicle routing algorithm for on-demand meal delivery using drones”. *Computers & Operations Research*, Vol. 111, PP. 1-20, 2019. [30](#)
- [151] Y. Marinakis and A. Migdalas. “Annotated Bibliography in Vehicle Routing,” *International Journal in Operational Research*, vol. 7, pp. 27-46, 2007. [13](#)
- [152] Y. Marinakis, A. Migdalas, P. Pardalos, “A new bilevel formulation for the vehicle routing problem and a solution method using a genetic algorithm”. *Journal of Global Optimization*, Vol. 38 (4), pp. 555-580, 2006. [39](#)
- [153] Y. Marinakis and M. Marinaki, “A hybrid genetic – particle swarm optimization algorithm for the vehicle routing problem”. *Expert Systems with Applications*, Vol. 37 (2), pp. 1446–1455, 2010. [39](#)
- [154] Y. Marinakis and M. Marinaki, “Bumble bees mating optimization algorithm for the vehicle routing problem”. In: *Handbook of Swarm Intelligence*. Springer, Berlin, Heidelberg, pp. 347–369, 2011. [39](#)

-
- [155] Y. Nagata and O. Bräysy, “Edge assembly based memetic algorithm for the capacitated vehicle routing problem”. *Networks*, Vol. 54 (4), pp. 205–215, 2009.
- [156] Z. Chen, H. Xu, “Dynamic column generation for dynamic vehicle routing with time windows”. *Transportation Science*, Vol. 40 (1), pp. 74–88, 2006.