國 立 成 功 大 學

工 業 與 資 訊 管 理 研 究 所

碩 士 論 文

使用智能櫃進行都會區最初與最終一哩收送與轉運物流之眾包運送問題研究

A first and last mile crowdshipping problem by smart lockers in city logistics

研 究 生： 李寶玉 Ananya Ariyarit

指導教授： 王逸琳

中 華 民 國 一 百 零 八 年 五 月

國立成功大學

碩士論文

使用智能櫃進行都會區最初與最終一哩收送與轉運物流之眾包運送問題研究

A first and last mile crowdshipping problem by smart lockers in city logistics

國立成功大學

研究生：李寶玉

本論文業經審查及口試合格特此證明

論文考試委員：王逸琳

莊坤達

陳[illegible]

吳政[illegible]

指導教授：王逸琳

系(所)主管：王惠嘉

中華民國 108 年 5 月 26 日

摘要

近年來電子商務演變及發展快速，線上購物風行，大量的小型貨物和運送宅配作業必須快速地於都會區中及時運送，已逐漸變成城市物流必須面對的重大挑戰。這些小包裹物流配送問題實為一個「最初與最終一哩收送問題」，其運送流程中遇到的最大困難在於顧客經常無法在物流人員抵達時在場收送件，因而導致物流人員可能要多次造訪而提高了物流成本。因應此困難，本論文提出一個嶄新的收送機制：透過於高交通流量區域設置智能櫃，除可方便顧客依其方便之時刻自行前往智能櫃收送件外，更可讓智能櫃擔任多段接力配送的最佳「轉運點」；亦即可將某些貨物分段運送，每段皆以智能櫃當轉運或收送點，供不同的運送人員以接力方式將貨物輾轉收送。智能櫃之轉運功能至今為止雖仍未見於文獻或實務場域中，然而我們認為該功能可與近年來逐漸風行的眾包運送機制結合，讓物流公司支付一些在外通勤的眾包司機，鼓勵其利用閒置時間與車內空間來協助收送貨，讓其在原先個人的途程規劃中順道協助收送件，以降低整體運輸成本與人力負擔，此種機制將可減少人力與無效的配送，應可有效地處理原先城市物流中因難的最初與最終一哩收送問題。

在本研究中，我們針對此種多段接力或收送過程中所必須處理的司機匹配問題(multi-hop rider matching problem) ，建構兩種整數線性規劃模型：以節線為基礎的數學模型(Arc-Based model)，以及時空網路模型(Time-Space Network model)。兩者皆已考量節點上的智能櫃轉運可能性，選取合適的眾包司機，並以最小的額外繞路成本完成所有收送任務為目標。由於這些數學規劃模式之求解過程相當耗時，為了加速求解過程，我們提出了一種傾向直送方式的貪婪式演算法(Greedy algorithm)；此外，再提出一個滾動式時窗演算法(Rolling horizon algorithm)，先將整體的規劃期間離散化，切割成多個較小的求解時段，再依序求解之。在進行數學模式及演算法之數值測試後，我們發現貪婪式演算法的求解效果雖較差，但其初始解可被用亦幫助縮短整數規劃模型的求解時間，且此種方法僅能求解中等規模的網路圖；反之，滾動式時窗演算法因為每回合僅求解當下及較近的未來時段內之最佳運送規劃，不用一次將全天的時段列入考慮，因此針對更大規模的網路亦可有不錯的求解表現。

關鍵字：眾包運送，最初與最終一哩收送，智能櫃，司機匹配問題，整數線性規劃

Abstract

As e-commerce grows and evolves, a large volume of small freights and home deliveries need to be handled every day in city logistics. A major challenging city logistics management problem for the couriers is to deal with failed parcel collection or deliveries. This is the “first and last mile delivery problem” and may very likely lead to multiple ineffective receiving and delivery attempts with high logistics costs. Here we propose a new shipping framework that utilizes pickup and delivery via smart lockers typically located in high traffic areas. This network of smart lockers can serve as convenient access points to remedy the first and last mile delivery problem.

In addition to serving the purpose for directly delivering or receiving goods, smart lockers can also be used for transhipment, which in fact has almost been ignored in literature and in practice. This function of transhipment will be very useful for crowdshipping companies, since thousands of crowdsourced drivers are commuting between home and businesses with spare space in their cars, in order to reduce shipping costs and effort, shipping companies are also considering paying these independent drivers to deliver parcels for them on the way to their destinations.

In this research, we investigate the multi-hop rider matching problem which takes the transhipment on nodes of smart lockers into consideration. The objective of this problem is to minimize the total cost of delivering all the parcels under consideration on time. We proposed two integer linear programming models: the Arc-based model, and the Time-space network model. Since these mathematical programming models are too time-consuming, we propose a greedy algorithm focusing more on better directing shipping. We also provide a rolling horizon algorithm to split the planning horizon into multiple smaller time periods and solve them sequentially. From our computational experiments the results indicate that the initial solution obtained by our greedy algorithm does help to shorten the solution time of the integer programming models to some extent for solving cases of medium scale, and the rolling horizon algorithm can help solve networks of larger scale in a shorter time.

Keywords: Crowdshipping, First and last mile delivery problem**, S**mart lockers, Rider matching problem, Integer linear programming

Acknowledgements

First, I would like to express my special appreciation and thanks to my advisor Prof. I-Lin Wang (王逸琳). It has been a good journey for me to live in Taiwan, got accepted in his lab, and under his advisee. I would like to thank you for encouraging my research and pushing me to exceed my limitability into another level. Your advice has been invaluable. I also thank all the committee members, Prof. Yeu-Shiang Huang (黃宇翔), Prof. Shine-Der Lee (李賢得), Prof. Kun-Ta Chuang (莊坤達), Prof. Sheng-I Chen (陳勝一) and Assoc. Prof. Cheng-Han Wu (吳政翰) for the suggestions and comments that make this thesis better.

Second, I would like to thank all I-Lin Lab member. To my seniors, BK, 思涵 and 冠緯, for being a good senior me and all of their patience helping me with a lot of things. To my classmate, Meidy, 嘉豪, Garry (昀軒) and 雪湄, for helping me and keep encourage me all the time. To my juniors, 晏慈, 彥瑋, 宗瀚 and Ari, thank you for all the support and sweet memories that you gave. Thank you, my Thai friends who always cheering me up, sending all the pictures and messages to me. My two years in NCKU will never be complete without all of them.

Last, I would love to express my thank you to my beloved family who supports me for everything. I can't thank you enough for encouraging me throughout this experience. My life in Taiwan is fulfilled because of always having them beside me at home. Thank you to all.

Ananya Ariyarit

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Background and motivation

According to Pitney Bowes annual Parcel Shipping Index (2017), the worldwide parcel volumes rose by 17% to 74.4 billion parcels in 2017, up from 63.6 billion in 2016, and they are expected to surpass the 100 billion mark in 2020. The increase in 2017 was in keeping with the 17-28% growth projection range predicted in studies the previous year. Generally, 22 parcels per person were shipped globally and 2,300 parcels were shipped every second in 2017. Therefore, the demand for courier services has risen over time. Because of this, companies have been looking new, efficient techniques for managing courier service operations with the purpose of gaining competence in the industry, maintaining high-quality service, reducing operating costs, and retaining and acquiring customers.

Corresponding to the growing volume of delivered parcels, increased customer expectations, and toughening market competition, retailers and logistics service providers are exploring and implementing innovative tools such as self-service technologies (SSTs). In the last mile delivery context, SSTs are presented in the form of smart lockers, which have advantages including being a simple, unstaffed delivery option (Faugere & Montreuil, 2017). Smart locker banks grouped into an unattended set of pickup and delivery lockers are a promising solution for last-mile parcel delivery, with a focus on unsuccessful deliveries and consolidation opportunities. Certainly, locker pickup is a convenient pickup option for consumers, while potentially driving delivery costs down by reducing the number of delivery points and avoiding unsuccessful deliveries that lead to multiple delivery attempts. The growing interest in smart lockers can be explained in part by the efficiency of algorithms that can be applied to today's technology in both sides of the business-to-customer (B2C) interaction, where the customer plays a significant the role as the service conductor.

A smart locker is a group of lockers, sited in apartment blocks, workplaces, railway stations, etc. The lockers have electronic locks with various opening codes, so they can be used by different consumers conveniently. Automated and equipped with interactive modules, they allow pickups and deliveries to occur in a few minutes. This solution is now appearing globally and has already been proven successful in European and Asian markets as a cheaper alternative to home delivery. Even though many courier companies have deployed such networks in the last decade, we believe that they have not adapted their distribution process to exploit the potential of these networks.

Apart from lockers, several service providers have recognized the opportunity that crowd shipment of parcels may offer. This crowdsourcing business option has a potential cost advantage because thousands of drivers are commuting between home and businesses with spare space in their cars, and these drivers pay for their own cars, gas, insurance, and maintenance costs. This also creates opportunities for faster deliveries and thus improves customer satisfaction. Traditionally, for a shipping company's B2C model to be profitable, a critical mass of customers has to be engaged for the provision of the service. Considering a crowd as a potential means to accomplish this goal, the time and effort required for arranging economically sustainable delivery may be substantially less. Up to the present time, crowd logistics has been seen mostly as an opportunity to reduce the cost and the speed of delivery in urban distribution of goods, in particular for home deliveries. Most of the literature on crowd logistics is limited to urban distribution and last mile activities, where crowd logistics has been considered intrinsic to city logistics.

However, shipping through crowdsourcing has some limitations. For example, if the route assigned to a driver adds excessive additional distance to the original route, it is highly likely that the order will be rejected. Due to this limitation, Herbawi and Weber (2012) provided a parameter to limit additional distance for the crowds engaged in the delivery process. However, limiting additional distance causes problems, because only one person may not be able to send the parcel directly from the origin to the destination, which makes the transfer of the parcel to a network of lockers a viable option, or more than one person may be required to deliver it.

These two solutions individually have already been shown to be successful to reduce the cost of distribution for both customers and service providers. However, in order to integrate these two solutions by using crowdsourcing to ship parcels via a network of lockers, our problem will be how to plan trips for the individuals delivering the parcels, where all drivers and parcels have their own origin, destination, earliest departure time, and latest arrival time. Our research focuses on the multi-hop rider matching problem, where the time window is emphasized.

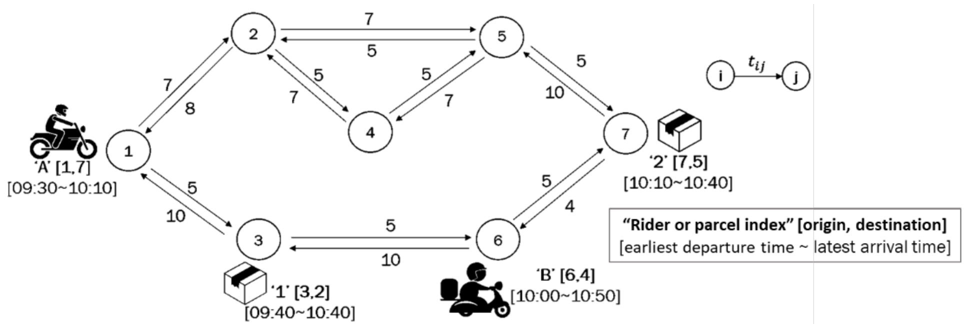

Figure 1.1 An illustrative example of the problem

The following examples illustrate the difficulty of decision making in this research question. Figure 1.1 shows the information for two drivers (A and B) and two parcels (1 and 2) in the network, with 7 stations, where driver A has to travel from station 1 to station 7, and driver B has to travel from Station 6 to Station 4. Parcel 1 has to be delivered from Station 3 to Station 2, and parcel 2 has to deliver from Station 7 to Station 5. Both drivers and parcels have specific earliest departure times and the latest arrival times. Also, the travel time to each station is shown in the network.

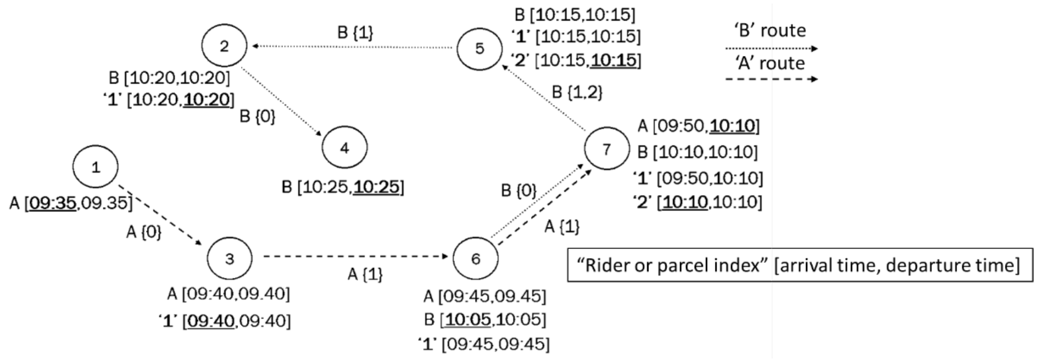

Figure 1.2 A solution for drivers A and B, when the station does not have a locker.

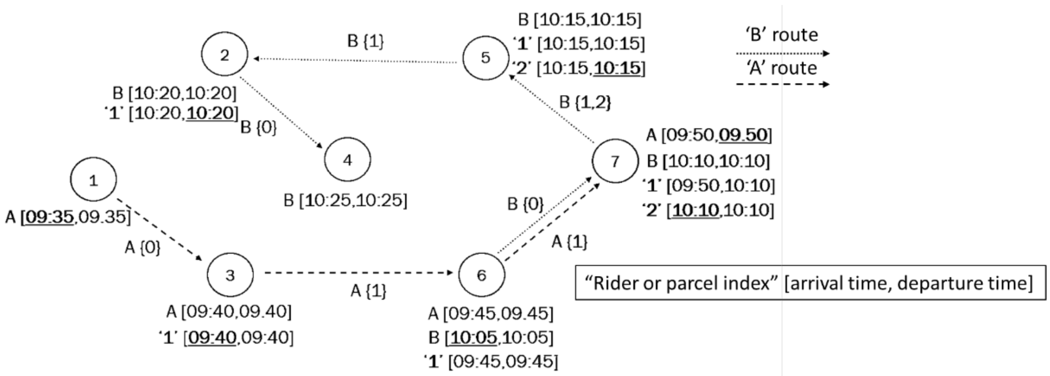

Figure 1.3 A solution for drivers A and B, when every station has a group of lockers

In Figure 1.2 and Figure 1.3, the solutions are given by the broken line and dotted line, with the broken line representing the route of driver A, and the dotted line representing the route of driver B. Figure 1.2 is a solution for a case where the station does not have a locker, which means that time synchronization has to occur to transfer parcel between drivers. Figure 1.3 is a solution for a case where every station has lockers. We showed with each arc in the solution the request carried by the driver. For example, in Figure 1.2, driver A carries parcel 1 from Station 3 to Station 6. We used “0” when the vehicle is not carrying any requests. At each station, we also showed the arrival and departure time for each driver. For example, in Figure 1.2 at Station 7 driver A arrives at 09:50 and departs at 10:10. The time the driver arrives and leaves the network is underlined. For example, in Figure 1.2 at Station 1, driver A arrives at the network at 09:35 and leaves the network from Station 7 at 10:05.

The difference between the solutions when there are no lockers and when there are lockers is shown at Station 7. When there is no locker, driver A has to wait until driver B arrives at Station 7, but when there are lockers, driver A can simply put the parcel in the locker and leave the network. This reduces the waiting time for driver A from 20 minutes to 0 minutes, which means the shipping company can reduce the cost for driver A. Table 1.1 provides a summary of the waiting time, the total time, and the shortest time.

Table 1.1 Summary of the waiting time, the total time and the shortest time

Case	**Does not have lockers**		**Has a group of lockers**		**Shortest time**
Driver	**Waiting time**	**Total time**	**Waiting time**	**Total time**	
A	20	35	0	15	15
B	0	25	0	25	22

However, the above example only considers the time constraints, but does not consider the limitation for additional distance for the driver. This issue makes the problem more complex.

1.2 Objectives

The goal of this work is to provide the means for a shipping company to match its demand for freight transportation with transport by people, with a specific focus on using the spare capacities of existing private vehicle flows with the objective to minimize the total cost of delivering all parcels on time.

To achieve this goal, we proposed a model that is based on matching and scheduling problem. This model can determine (1) the optimal matching plan between drivers and parcels for the whole planning horizon (e.g., one day), (2) the optimal path of each driver and each parcel, and (3) the time schedule for the drivers and the parcels to be delivered by independent drivers.

1.3 Scope and limitation

Considering a combination of the existing planned routes of the drivers, we limit our attention to the problem offline: given all the drivers and the known delivery requests (i.e. origin, destination, earliest departure time, and latest arrival time), find an optimal plan to deliver all the parcels on time, while ignoring possible future requests. In contrast to P2P platforms where users generally expect a direct response, we focus on periodic planning to benefit from the consolidation of resources, which makes sense from the perspective of a shipper. The offline configuration allows us to group incoming requests intelligently and facilitates multi-driver multi-parcel matching. Even a driver with a completely different destination can take a parcel to an intersection where the parcel can be transferred to other vehicles that travel closer to the destination.

1.4 Overview of thesis

The rest of this thesis organized as follows: Chapter 1 presents an introduction, including background and motivation, research objectives, scope and limitations, and an overview of the thesis. Chapter 2 reviews the locker network, multi-hop rider matching problems, and related studies. Chapter 3 describes the research approach including the problem description and provides a mathematical model to solve the problem. In Chapter 4, a greedy heuristic is proposed to solve the large problem. In Chapter 5, we present the results of the numerical experiments. Finally, we provide a conclusion and recommendations for future research in Chapter 6.

Chapter 2 Literature Review

In this chapter, we mention some literature related to our research. Our research is related to two main topics, the first of which is the practical characteristics of a locker network and the theoretical research, and the second of which concerns variations in the ride sharing problem and related research as compared to the current study.

2.1 Lockers network

The supply chain industry is moving very fast, and new generation last mile delivery solutions are already being tested in many places around the world. The use of lockers is one of the solutions that is globally emerging and has already been proven successful in European and Asian markets as a cheaper alternative to home delivery (Faugere & Montreuil, 2016). Smart locker banks currently operate in many countries such as Germany, Singapore, Poland, and China, among others.

Locker terminals can be compared to business partners serving as consolidation delivery points, also called service points. In certain markets, last mile delivery networks are often composed of both locker terminals and business partners, for the purpose of creating a balance between automation and human interaction as well as dealing with increasing investments in equipment and real estate while growing the network.

Locker Terminals: Networks composed of smart locker terminals are used to accommodate the population by being close enough to the last customers and to assure that it is worth choosing smart locker terminal delivery over regular home delivery. The terminals usually are located in high traffic areas such as train stations, malls, or public spaces. According to the different storage capacity and layouts available, they can be adapted to every location and any level of demand. Faugere and Montreuil (2017) listed the advantages and disadvantages of four different lockers design alternatives, from the current solution of fixed-configuration lockers to innovative solutions comprising smart mobile modular lockers.

Service Points: Some logistics providers use access points along with locker terminals to build their network. Access points are local businesses that partner with a logistics provider to receive and store parcels during the hours of operations, until the final customer pick up the package. Although it requires a workforce (the staff of the local business) to operate and it limits the operating hours of the business, this solution makes it possible to build a large network with a minimum of investments in assets since it works through partnerships with existing facilities. This means that it is also more flexible in term of capacity and deployment, and as a partnership can be created or stopped if needed. Morganti et al. (2014) provided an in-depth look at alternative parcel delivery services in France and Germany that have been using service points as the pickup point.

Both of these solutions to building networks use consolidated delivery opportunities and allow logistics providers to save money and time in last mile delivery, while allowing more flexibility as to the time the packages get delivered. However, using both smart locker terminals and service points together seems to be the solution to build a large network with an acceptable level of investment (Faugere & Montreuil, 2016).

The use of lockers already was proven to be successful not only in a practical way but also in studies verifying this. Kämäräinen (2001) used delivery data from the suburban area of Helsinki to compare the costs of regular home delivery with delivery to a service point. The results showed a cost reduction of 42% when using the service point. Lemke, Iwan, and Korczak (2016) used data provided by a Polish postal service company to assess the use of parcel lockers in Poland, focusing on their current locations. The results showed that 15% of consumers would use the lockers more frequently if their locations were improved, and most consumers indicated a preference for lockers near their homes. Punakivi (2001) used sales data from a large retail company in Finland and compared the operating costs of regular home delivery with delivery to service points and lockers. The results showed that transportation costs when delivering to lockers were 55–66% lower than those when using attended receipt with a two-hour delivery time window.

However, a few related studies have been done on how to create the best design and operating strategies for locker networks and determining how to quantify their benefits. One of these was done by Deutsch and Golany (2017), who focused on choosing the optimal number, locations, and sizes of locker facilities given a specific area and specific customer demands. They indicated where to locate them and how many lockers to install at each location in order to maximize the total profit by presenting a binary integer linear program as a statistical approach.

2.2 Multi-hop rider matching problem

The multi-hop rider matching problem can be formulated as a special case of the general pick-up and delivery problem (GPDP). GPDP consists of devising a set of routes to satisfy transportation requests with given loads and origin/destination locations. Vehicles that operate these routes have a specific origin, destination, and capacity (Savelsbergh and Sol (1995)).

The dial-a-ride problem (DARP) is a special case of the GPDP, where all vehicles share the same origin and destination depot and transport the loads to people. Although DARP is usually used in systems that aim at transporting elderly or handicapped people, this problem is very close to the rider-matching problem in ridesharing systems.

The rider-matching problem has attracted attention in academia particularly in recent years. Rider-matching problems share some characteristics with more advanced DARPs, such as multiple depots and heterogeneous vehicles and passengers. Drivers in ridesharing systems are traveling to perform activities and have distinct origin and destination locations (multi-depot), different vehicle capacities (heterogeneity), and rather narrow travel time windows. These factors can lead to the matching problems in ridesharing systems being generally spatiotemporally sparse. One characteristic that differentiates the rider-matching problem from DARP is the fact that the set of vehicles in a ridesharing system is neither fixed (i.e., no fleet sizes are available on a regular basis) nor deterministic (i.e., the system does not know in advance the time windows and origins and destinations of drivers). In addition, drivers who make their vehicles available in ridesharing systems are peers to the passengers who are looking for rides, and therefore, service quality measurement reserved only for passengers in DARPs should be extended to drivers as well in ridesharing systems.

Agatz et al. (2012) and Furuhata et al. (2013) classify ride-sharing systems based on different criteria and discuss the challenges ridesharing systems face. It clarifies that the rider-matching problem matching each driver with a single rider. This can be modeled as a maximum-weight bipartite matching problem that minimizes the total rideshare cost (Agatz et al., 2011).

Table 2.1 Ride-Share Variants **(**Agatz et al., 2012)

	Single rider	**Multiple riders**
Single driver	Matching a pair of drivers and riders	Routing of drivers to pick up and deliver riders
Multiple drivers	Routing of riders to transfer between drivers	Routing of riders and drivers

Optimally, matching drivers offering a ride with riders requesting a ride is easy for the static variant in which a single driver takes along a single rider. This variant is simple because there are a polynomial number of potential matches, and determining the optimal route sequence for a given potential match is simple. In all other variants, determining the best route sequence for a given match, which may involve multiple drivers and riders, can be more complicated (Agatz et al. ,2012).

In its basic form, DARP is considered to be a depot where a fleet of homogeneous vehicles start their trips in the morning and to which they return at the end of their shifts. Each passenger is assumed to make the entire trip in the same vehicle; i.e. the possibility of transfers between vehicles is not considered. Variants of DARP that are more application-friendly consider time windows for the pick-up and delivery of passengers. Cordeau and Laporte (2007) provide an overview of the literature on DARP.

In reality, the problem of transporting passengers is often more complex than the basic form of DARP. Some agencies have their fleet located at stations throughout their operating area. This motivated the development of the multi-depot formulation for DARP (MD-DARP) (Cordeau and Laporte, 2007). Recently, an additional degree of flexibility was added to the original DARP that allows the possibility for passengers to transfer between multiple vehicles/modes of transportation, leading to the emergence of the DARP with transfers (DARPT). Masson et al. (2014), Stein (1978), and Liaw et al. (1996) are the only researchers that have studied this variant of the problem, to the best of our knowledge. Masson et al. (2014) limited the number of potential transfers to one. Stein (1978) did not put a constraint on the capacity of vehicles, and worked with demand at an aggregate level, rather than the individual passengers' travel desires. Liaw et al. (1996) use heuristic algorithms to propose multi-modal routes to para-transit users. In their study, they attempted to route para-transit vehicles to carry passengers from their homes to bus stops and from bus stops to their destinations.

There are also rider-matching problems that are more complex that try to take advantage of the full unused capacity of vehicles by allowing multiple riders in each vehicle. This form of ridesharing is similar to the carpooling problem where a large employer encourages its employees to share rides to and from work (Baldacci et al., 2004 and Wolfler Calvo et al., 2004). The taxi-sharing problem, as formulated by Hosni et al. (2014), also is an attempt to reduce the cost of taxi services by having people share their rides. Herbawi and Weber (2012) have studied the problem of matching one driver with multiple riders in the context of ridesharing and proposed non-exact evolutionary multi-objective algorithms. Stiglic et al. (2015) managed to increase the number of served riders by having riders walk to meeting points, where multiple riders could be picked up by a driver. The number of stops for each driver, however, was limited to a maximum of two.

Herbawi and Weber (2011a, b) modeled another variant of the rider-matching problem in which a single rider could travel by transferring between multiple drivers. They proposed a genetic algorithm to solve this many-to-one matching problem. Masson et al. (2015) studied a similar problem in a multi-modal environment, where goods were carried using a combination of excess bus capacities and city freighters. They proposed an adaptive large neighborhood heuristic algorithm to solve the problem. Coltin and Veloso (2014) showed the fuel efficiency that could be gained by including transfers in the riders' itineraries using three heuristic algorithms.

Many-to-many matching problems allow drivers to have multiple passengers on board at each point in time and also allow riders to transfer between drivers (Agatz et al., 2010). Cortés et al. (2010) were the first to formally formulate a many-to-many pick-up and delivery problem. They introduced an exact branch-and-cut solution method. The largest example that they solved only consisted of 6 requests, two vehicles, and one transfer point.

To the best of our know arc, there are limited studies that model many-to-many ridesharing systems. Agatz et al. (2012) was one of the first efforts to take an optimization approach toward modeling many-to-many ridesharing systems. In their study, the authors discuss modeling multi-modal ridesharing systems that allow for transfers between different modes of transportation. However, they did not discuss a solution methodology. Arslan et al.

(2016) considered a new dynamic variant of the pickup-and-delivery problem with time windows where some of the tasks could be matched to a given set of ad-hoc drivers while a third-party backup fleet completed the remaining tasks. An ad-hoc driver could stop multiple times along the route to pick up and deliver several parcels, and each parcel was picked up at its origin and dropped off at its ultimate destination. A static version of the problem was formulated as a matching problem with side constraints. The dynamic version was heuristically solved by solving the static problem with a rolling horizon.

Chen et al. (2017) suggested using a method to harness the spare capacity of journeys made by private vehicles to deliver parcels. These scholars developed a door-to-door delivery scheme, where parcels are assigned to planned trips with the possibility of a small detour. Parcels could be delivered by more than one driver, but the transfers between drivers required time synchronization. The multi-driver multi-parcel matching problem was defined as the problem of assigning parcels to drivers and determining the transfer point that would minimize the total cost. Parcels could either be delivered by drivers or by a third-party courier company at a given, higher cost. The authors formulated a static version of the problem as an integer linear programming (ILP) problem and proposed a heuristic algorithm that could be used to solve a dynamic version of the problem.

The most similar to our research was a study by Chen et al. (2017) since in their problem, the transfers between driver required time synchronization; i.e., if the driver already delivered the parcel to some specific location that was not the destination for this parcel, and the next driver intended to pick this parcel had not arrived at this location, the first driver had to wait until the next driver arrived. However, in our work, we add lockers to the network, so the transfers between drivers do not require time synchronization if there is a locker at that location. We consider a problem with (1) multiple drivers, (2) multiple parcels, (3) time windows, (4) the routing of the drivers, (5) multiple hops of the parcel, where (6) the locker comprises the pickup, delivery, and transfer nodes.

Table 2.2 provides a comparison between our research and related studies.

Table 2.2 Research Comparison

Reference	**Multi parcels**	**Multi drivers**	**Time window**	**Allow detour**	**Allow transfer**	**Locker network**
Herbawi and Weber (2011)	-	✓	✓	-	✓	-
Agatz et al. (2012)	✓	-	✓	✓	-	-
Herbawi and Weber (2012)	✓	✓	✓	✓	-	-
Arslan et al. (2016)	✓	✓	✓	✓	-	-
Chen et al. (2017)	✓	✓	✓	✓	✓	-
Our thesis	✓	✓	✓	✓	✓	✓

Chapter 3 Research Approach

In this chapter introduces modeling of multi-hop rider matching problem. In order to obtain the exact solution, we propose a two integer linear programming model base on the arc-based model and the time-space network.

3.1 Problem Description

As e-commerce has grown and evolved, shipping companies have increasingly had deliver large volumes of small freight and home deliveries every day, so they have developed a new delivery variation consisting of pickup and delivery via smart lockers located in high traffic areas. The customers can come to pick whenever they want, and deliveries can be made at any time. This reduces costs by reducing the number of delivery points and avoiding unsuccessful deliveries that result in multiple delivery attempts.

Since thousands of drivers are commuting between their homes and businesses with spare space in their cars, to reduce shipping costs and effort, shipping companies also consider paying independent drivers to deliver the parcels for them on the way to their destinations. To accommodate the parcel delivery, the driver may have to detour or make extra stops. The length of the detour and the number of extra stops is determined by the driver's willingness to extend the trip with respect to both distance and time. Drivers may take a single parcel or multiple parcels (sequentially or simultaneously) along their journey, as long as the capacity of their vehicle is not exceeded. Similarly, parcels may be carried by a single driver from their origins to their destinations or may be transported by multiple drivers and transferred from one to another en route to their destinations. All of the activity mentioned above is moving in a road network, whether it is to pick up, deliver, or transfer to a locker.

We consider a directed network $G = (N, A)$, where N is the set of nodes representing the possible locations for departure, arrival or transfer, and A is the set of arcs that represent the road network. A distance d_{ij} and travel time e_{ij} are associated with each arc $(i, j) \in A$. In addition, we are given a set of drivers L and a set of parcels K. Driver $l \in L$ will travel from his origin O_l^L to his destination W_l^L. The earliest time E_l^L at which he can depart from his origin O_l^L and the latest time G_l^L at which he has to arrive at his destination W_l^L are associated with driver l. Driver l has V_l spare space available for parcels. Similarly, each parcel $k \in K$ will travel from its origin O_k^K to its destination W_k^K. The earliest time E_k^K at which it can depart from its origin O_k^K and the latest time G_k^K at which it has to arrive at its destination W_k^K are also associated with parcel k. Each node i has a limited available number of lockers R_i.

To deal with realistic requirements, our model has the following features: First, drivers are allowed to deviate from their shortest path to pick up and drop off parcels, as long as their detour is at most a fraction δ of their shortest path length, and thus, the routing of the drivers also has to be considered. Second, parcels are not as time sensitive as riders in the ride sharing problem, as long as they are delivered within the associated time windows. To avoid making too many unnecessary transfers, parcels are not allowed to pass the same node more than once in our model. To avoid departing from their shortest path too much, drivers are also not allowed to visit the same node twice.

3.2 Problem assumption

In order to properly simplify the mathematical model and the limitations of the use of the model, the following basic assumptions are made:

1. We assume the size of all the lockers is identical while ignoring the size of the parcel.
2. The origin, destination, earliest departure time and latest arrival time for all the parcels and drivers are known beforehand.
3. The traveling times are known and constant.
4. We ignore possible future requests.
5. The original route for all drivers is the shortest path from their origin to their destination
6. Each parcel is not allowed to pass the same node more than once.
7. Each driver is also not allowed to visit the same node twice

3.3 A proposed mathematical formulation

In this section, we develop two mathematical models for our problem from a shipping company's perspective based on the arc-base model and a time-space network

3.3.1 Notation

The following notation used in the mathematical model is listed as follows:

Set

L	Set of drivers
K	Set of parcels
N	Set of nodes
A	Set of arcs

Parameters

T	Time period, $t = 1,...,T$
O_l^L	Driver l's origin
W_l^L	Driver l's destination
O_k^K	Parcel k's origin
W_k^K	Parcel k's destination
E_l^L	Earliest departure time for driver l
G_l^L	Latest arrival time of driver l
s_l	Distance of the shortest path from O_l^L to the W_l^L of driver l
δ	Coefficient of maximum detour
E_k^K	Earliest departure time for parcel k
G_k^K	Latest arrival time for parcel k
V_l	Available car capacity of driver l
R_i	Available locker storage capacity of node i
d_{ij}	Travel distance from node i to node j; $\forall i, j \in N$
e_{ij}	Travel time from node i to node j; $\forall i, j \in N$

c_k	Cost of delivering parcel k by the shipping company
w_1	Cost per parcel per kilometer for a driver who helps carry the parcel
w_2	Cost of transferring a parcel between drivers
w_3	Cost per minute for a driver waiting on the way
c_p	Penalty cost per kilometer if the driver has additional travel distance more than limited detour
M	Large number

3.3.2 Arc-based Model M^A

The arc-based model is based on Chen et al.'s (2017) formulation. Using the decision variables and dependent variables, the notation uses for the arc-based model is as follows:

Decision variables

Z_{ij}^{l}	Binary variable equal to 1 if driver l goes directly from node i to node j; and 0 otherwise
X_{ij}^{kl}	Binary variable equal to 1 if driver l carries parcel k from node i to node j; and 0 otherwise
C_k	Binary variable equal to 1 if parcel k delivered by shipping company
β_{li}^{L}	Departure time of driver l at node i
β_{ki}^{K}	Departure time of parcel k at node i

Dependent variables

S_i^{kl}	Binary variable equal to 1 if driver l picks up parcel k at node i
α_{li}^{L}	Arrival time of driver l at node i
α_{ki}^{K}	Arrival time of parcel k at node i

The objective of this model is to minimize the overall cost of the shipping company related to the parcel delivery service, which consists of the shipping cost incurred from self-delivery, the three weighted costs of reward the driver and one penalty cost. The three weight costs include (1) the transportation cost calculated by the kilometers that the drivers travel with parcels, (2) the risk and inconvenience associated with the number of parcel transfers and (3) the waiting time for transferring parcels. The penalty cost is for limited additional distance. The objective function is shown as Equation (3.1).

$$\begin{aligned} Min \quad & \sum_k c_k C_k + w_1 \sum_k \sum_l \sum_{i,j} d_{ij} X_{ij}^{kl} + w_2 \sum_k \sum_l \sum_{i \neq O_k^K} S_i^{kl} \\ & + w_3 \sum_l \left(\left(\alpha_{l,W_l^L}^L - \beta_{l,O_l^L}^L \right) - \sum_{i,j} e_{ij} Z_{ij}^l \right) \\ & + c_p \sum_l \left(\sum_{i,j} \sum_t d_{ij} Z_{ij}^l - s_l (1+\delta) \right) \end{aligned} \tag{3.1}$$

This model is confined by two sets of constraints: (1) space network constraints and (2) capacity and time constraints

Space network constraints

$$\sum_j Z_{ij}^l = 1 \qquad \forall l, i = O_l^L \tag{3.2}$$

$$\sum_i Z_{ij}^l - \sum_{i'} Z_{ji'}^l = 0 \qquad \forall l, \forall j \in N \setminus \{O_l^L, W_l^L\}, i \neq i' \tag{3.3}$$

Constraints (3.2) and (3.3) ensure that each driver will take only one path and that this path is continuous.

$$\sum_i Z_{ij}^l = 0 \qquad \forall l, j = O_l^L \tag{3.4}$$

$$\sum_i Z_{ij}^l \leq 1 \qquad \forall l, j \tag{3.5}$$

Constraints (3.4) and (3.5) ensure that each driver can visit each node at most once.

$$\sum_{i,j} e_{ij} Z_{ij}^l \leq G_l^L - E_l^L \qquad \forall l \tag{3.6}$$

Constraints (3.6) limit the traveling time for each driver.

$$\sum_l \sum_j X_{ij}^{kl} + C_k = 1 \qquad \forall k, i = O_k^K \tag{3.7}$$

$$\sum_i \sum_l X_{ij}^{kl} - \sum_l \sum_{i'} X_{ji'}^{kl} = 0 \qquad \forall k, \forall j \in N \setminus \{O_k^K, W_k^K\}, i \neq i' \tag{3.8}$$

Constraints (3.7) and (3.8) ensure that each parcel must be deliver from its origin to its destination either by a driver or by a shipping company.

$$\sum_l \sum_i X_{ij}^{kl} = 0 \qquad \forall k, j = O_k^K \qquad (3.9)$$

Constraint (3.9) ensures that the parcel will not return to its origin.

$$X_{ij}^{kl} \le Z_{ij}^l \qquad \forall k, l, i, j \qquad (3.10)$$

Constraint (3.10) ensures that the parcels cannot travel without a driver if the parcel is scheduled to be delivered by a driver.

$$S_j^{kl} \ge \sum_i X_{ji}^{kl} - \sum_i X_{ij}^{kl} \qquad \forall k, l, j \in N \setminus \{O_k^K, W_k^K\} \qquad (3.11)$$

Constraint (3.11) keeps track of the stations where the driver will leave the parcel.

$$Z_{ij}^l, X_{ij}^{kl}, C_k, S_i^{kl} \in \{0,1\} \qquad \forall k, l, i, j \qquad (3.12)$$

Constraint (3.12) is a domain constraint.

Constraints (3.2) – (3.12) are imposed to find feasible matches between drivers and parcels based on the space network information (i.e., origin and destination).

Capacity and time related constraints

$$\sum_k X_{ij}^{ki} \le V_l \qquad \forall l, i, j \qquad (3.13)$$

Constraint (3.13) is the capacity constraint for the drivers.

$$\sum_k \sum_l S_i^{kl} \le R_i \qquad \forall i \qquad (3.14)$$

Constraint (3.14) is the storage capacity constraint for the locker at each node.

$$\alpha_{lj}^L \ge \beta_{li}^L + e_{ij} - M(1 - Z_{ij}^l) \qquad \forall l, \forall i \in N \setminus \left\{W_l^L\right\}, \forall j \in N \setminus \left\{O_k^K\right\} \qquad (3.15)$$

Constraint (3.15) calculates the arrival time of drivers based on the associated departure and travel times.

$$\beta_{ki}^K \ge E_k^K (1 - C_k) \qquad i = O_k^K, \forall k \qquad (3.16)$$

$$\alpha_{ki}^K \le G_k^K (1 - C_k) \qquad i = W_k^K, \forall k \qquad (3.17)$$

Constraints (3.16) and (3.17) ensure that the parcel delivered by the driver must depart after the corresponding earliest departure time and arrive before the corresponding latest arrival time.

$$\beta_{ki}^{K} \geq \alpha_{ki}^{K} \qquad \forall i \in N \setminus \left\{O_k^K, W_k^K\right\}, \forall k \tag{3.18}$$

Constraint (3.18) ensures that the departure time for each parcel is not earlier than the arrival time at the same node.

$$\beta_{li}^{L} \geq E_l^L \qquad i = O_l^L, \forall l \tag{3.19}$$

$$\alpha_{li}^{L} \geq G_l^L \qquad i = W_l^L, \forall l \tag{3.20}$$

$$\beta_{li}^{L} \geq \alpha_{li}^{L} \qquad \forall i \in N \setminus \{O_l^L, W_l^L\}, \forall l \tag{3.21}$$

Constraints (3.19) – (3.21) are similar to Constraints (3.14) – (3.18), the time compatibility issues for the drivers. Constraints (3.19) and (3.20) ensure the earliest departure time and the latest arrival time at the origin and the destination for each driver. Constraint (3.21) ensures that the departure time must be later than arrival time at the same station for each driver.

$$\beta_{ki}^{K} - \beta_{li}^{L} \leq M\left(1 - \sum_j X_{ij}^{kl}\right) \qquad \forall k, l, \forall i \in N \setminus \left\{W_k^K, W_l^L\right\} \tag{3.22}$$

$$\beta_{ki}^{K} - \beta_{li}^{L} \geq -M\left(1 - \sum_j X_{ij}^{kl}\right) \qquad \forall k, l, \forall i \in N \setminus \left\{W_k^K, W_l^K\right\} \tag{3.23}$$

Constraints (3.22) and (3.23) ensure that the departure time of a parcel is equal to the departure time of the driver who will carry it.

$$\alpha_{li}^{L} - \alpha_{ki}^{K} \leq M\left(1 - \sum_j X_{ij}^{kl}\right) \qquad \forall k, l, \forall i \in N \setminus \left\{O_k^K, O_l^L\right\} \tag{3.24}$$

$$\alpha_{li}^{L} - \alpha_{ki}^{K} \geq -M\left(1 - \sum_j X_{ij}^{kl}\right) \qquad \forall p, q, \forall i \in N \setminus \left\{O_k^K, O_l^L\right\} \tag{3.25}$$

Constraints (3.24) and (3.25) guarantee that the arrival time of the parcel is equal to the arrival time of the driver who carries it. Thus, constraints (3.22) – (3.25) ensure the time consistency of a parcel and all the drivers carrying it.

$$\beta_{li}^{L} - \alpha_{ki}^{K} \geq -M\left(1 - \sum_j X_{ij}^{kl}\right) \qquad \forall k, l, \forall i \in N \setminus \left\{W_l^L, W_k^K\right\} \tag{3.26}$$

Constraint (3.26) guarantees that if the driver who will carry the parcel arrives earlier than the parcel, the driver will wait until the parcel arrives.

$$\beta_{li}^{L}, \beta_{ki}^{K}, \alpha_{li}^{L}, \alpha_{ki}^{K} \geq 0 \qquad \forall k, l, i \qquad (3.27)$$

Constraint (3.27) is a domain constraint.

In this model, storage capacity of the locker does not change based on time, means that the locker cannot be reused until the end of the time period by in the real world it's likening the end of the operation time or the end of the day. Thus, how to make it possible for the locker to be reused the same time period will be discussed in Section 3.3.3.

3.3.3 **Time-Space Network model** M_T^N

To make it possible for a locker to be reused on the same day, we use a time-space Network that can track the number of lockers available in each time period. By using this model, the original network (road network) has to be transformed into a time-space network. An example of the time-space network for the problem described above is shown in Figure 3.1

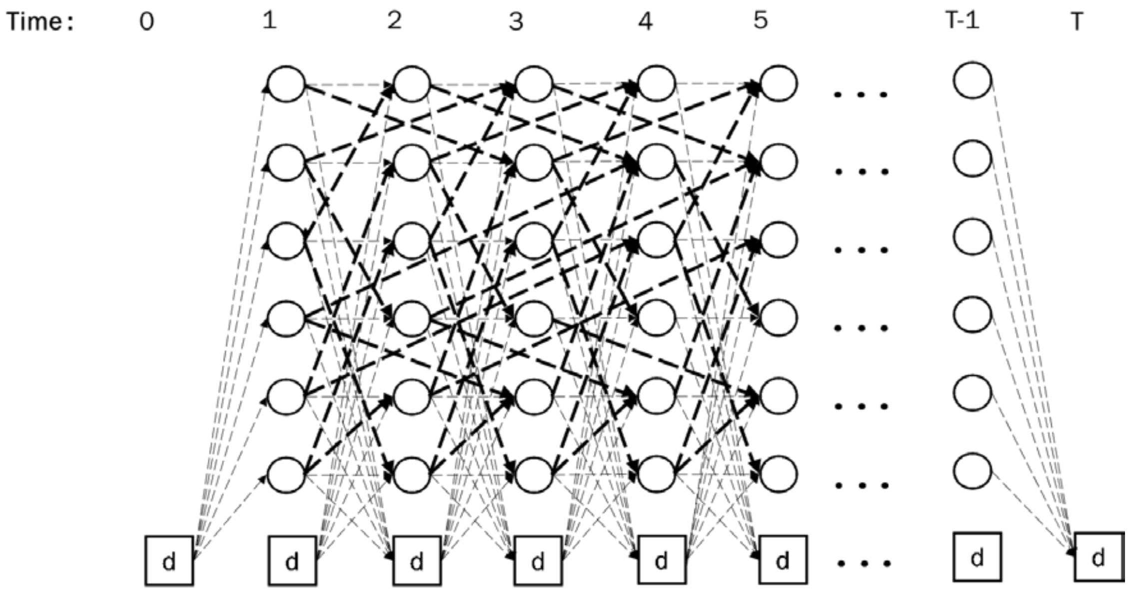

Figure 3.1 An illustrative example of a time-space network

In this network, once given the driver and parcel information, the driver can depart anytime from his origin after the earliest departure time by going from to his origin, he will appear in the network at his earliest departure time which he will depart from dummy node to his origin that takes one time period before their earliest departure time. After arriving at his destination, he will leave the network at the next time period to go to a dummy node. Similar which the driver the parcel to appear in the network at its earliest departure time, it is necessary to go from dummy node to its origin in one time period before its earliest departure time. Once it arrives at its destination, it leaves the network in the next time period by going to a dummy node. For the parcel, if its need to go to another node it must carry by some driver, but if it will stay at the same node in next time period it’s no need to have drivers carry its. The flow for driver and parcel described above is shown in Figure 3.2

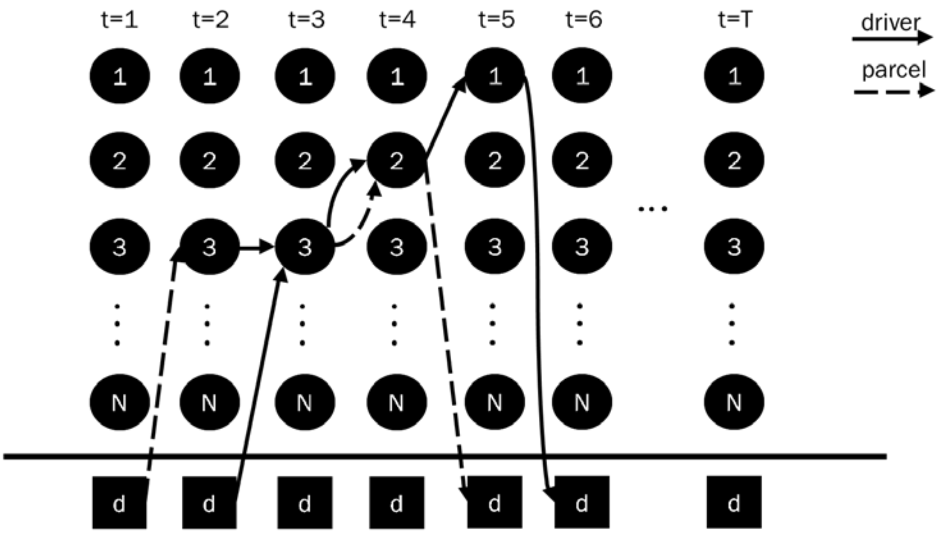

Figure 3.2 The flows and routings for driver and parcel in time-space network

The decision variables and dependent variables used for the time-space network model are as follows:

Decision variables

Z_{ij}^{lt} Binary variable equal to 1 if driver l goes directly from node i to node j at the end of time period t; and 0 otherwise

X_{ij}^{klt} Binary variable equal to 1 if driver l carries parcel k from node i to node j at the end of time period t; and 0 otherwise

Y_{ij}^{kt} Binary variable equal to 1 if parcel k goes directly from node i to node j at the end of time period t; and 0 otherwise

C_k Binary variable equal to 1 if parcel k is delivered by a shipping company

Dependent variables

S_i^{klt} Binary variable equal to 1 if driver l picks up parcel k from node i at the end of time period t

α_{li}^{L} Arrival time of driver l at node i

α_{ki}^{K} Arrival time of parcel k at node i

β_{li}^{L} Departure time of driver l at node i

β_{ki}^{K} Departure time of parcel k at node i

r_i^t Number of available lockers at node i at the end of time period t

The objective of this model is same as that of the arc-based model, which is to minimize the overall cost of the shipping company related to the parcel delivery service, which consists of the shipping cost incurred from self-delivery, the cost of rewards, which includes (1) the transportation cost, (2) the risk and inconvenience associated with the number of parcel transfers and (3) the waiting time for transferring parcels and the penalty cost is for limited additional distance. The objective function is shown as (3.28).

$$\begin{aligned} Min \quad & \sum_k c_k C_k + w_1 \sum_k \sum_l \sum_{i,j \in A; i \neq j} \sum_t d_{ij} X_{ij}^{klt} + w_2 \sum_k \sum_l \sum_{i \neq O_k^K} S_i^{kl} \\ & + w_3 \sum_l \left(\left(\alpha_{l,W_l^L}^{L} - \beta_{l,O_l^L}^{L} \right) - \sum_t \sum_{i,j \in A; i \neq j} e_{ij} Z_{ij}^{lt} \right) \\ & + c_p \sum_l \left(\sum_t \sum_{i,j \in A; i \neq j} d_{ij} Z_{ij}^{lt} - s_l (1+\delta) \right) \end{aligned} \tag{3.28}$$

This model is confined by two sets of constraints: (1) space network and time window constraints (2) and capacity constraints.

Space network and time window constraints

$$Z_{dj}^{lt} = 1 \qquad \forall l, j = O_l^L, t = E_l^L - 1 \tag{3.29}$$

Constraint (3.29) ensures that each driver can depart from their origin any time after the earliest departure time.

$$Z_{dj}^{lt} = 0 \qquad \forall l, j \in N \setminus \{O_l^L\}, t \qquad (3.30)$$

Constraint (3.30) ensures that each driver can only start at their origin.

$$\sum_{t=1}^{T} \sum_{i} Z_{ij}^{lt} \leq 1 \qquad \forall l, j \in N; j \neq i \qquad (3.31)$$

Constraint (3.31) guarantees that each driver can visit each node at most once.

$$\sum_{t=E_l^L}^{G_l^L} \sum_{i} Z_{ij}^{lt} = 1 \qquad \forall l, j = W_l^L \qquad (3.32)$$

Constraint (3.32) guarantees that each driver must arrival at their destination before their latest arrival time

$$\sum_{i} Z_{ij}^{l(t-e_{ij})} - \sum_{i'} Z_{ji'}^{lt} = 0 \qquad \forall l, j, t = E_l^L + 1, \ldots, G_l^L; t - e_{i,j} \geq E_l^L \qquad (3.33)$$

Constraint (3.33) shows the flow balance for the driver at each node

$$\sum_{j} Y_{ij}^{kt} + C_k = 1 \qquad \forall k, i = O_k^K, t = E_k^K \qquad (3.34)$$

Constraint (3.34) ensures that each parcel is delivered from its origin to its destination either by a driver or by a shipping company within the parcel time window.

$$\sum_{j} Y_{O_k^k j}^{kt} = \sum_{t'=E_k^K}^{G_k^K} \sum_{i} Y_{iW_k^K}^{kt'} \qquad \forall k, t = E_k^K \qquad (3.35)$$

Constraint (3.35) ensures that parcel must arrive at its destination if at its origin have pickup by the driver

$$Y_{dj}^{kt} = 1 \qquad \forall k, j = O_k^K, t = E_k^K \qquad (3.36)$$

Constraint (3.36) defines the earliest departure time for each parcel

$$\sum_{i} Y_{ij}^{k(t-e_{ij})} - \sum_{i} Y_{ji}^{kt} = 0 \qquad \forall k, j, t; (t - e_{ij}) \geq E_k^K \qquad (3.37)$$

Constraint (3.37) shows the flows balance for each parcel at each node each time period

$$Y_{ij}^{kt} \leq \sum_{l} Z_{ij}^{lt} \qquad \forall k, i, j, t; i \neq j \qquad (3.38)$$

Constraint (3.38) ensures that parcel cannot go to another node without the driver

$$2X_{ij}^{klt} \le Y_{ij}^{kt} + Z_{ij}^{kt} \qquad \forall k,l,t,i \ne j \tag{3.39}$$

$$X_{ij}^{klt} \ge Y_{ij}^{kt} + Z_{ij}^{kt} - 1 \qquad \forall k,l,t,i \ne j \tag{3.40}$$

Constraints (3.39) - (3.40) specify the driver who carries the parcel to another node

$$\beta_{ij}^{L} = \sum_t t \cdot Z_{ij}^{lt} \qquad \forall l,i \ne j \tag{3.41}$$

$$\alpha_{lj}^{L} \ge \beta_{li}^{L} + e_{ij} - M(1 - \sum_t Z_{ij}^{lt}) \qquad \forall l,i,j \tag{3.42}$$

Constraints (3.41) and (3.42) calculate the departure time and arrival time for each driver at each node

$$\beta_{kj}^{K} = \sum_t t \cdot Y_{ij}^{kt} \qquad \forall k,i \ne j \tag{3.43}$$

$$\alpha_{kj}^{K} \ge \beta_{ki}^{K} + e_{ij} - M(1 - \sum_t Y_{ij}^{kt}) \qquad \forall k,i,j \tag{3.44}$$

Constraints (3.43) and (3.44) calculate the departure time and arrival time for each parcel at each node

$$Z_{ij}^{lt}, X_{ij}^{klt}, Y_{ij}^{kt}, C_k, S_i^{klt} \in \{0,1\} \qquad \forall k,l,i,j \tag{3.45}$$

$$\beta_{li}^{L}, \beta_{ki}^{K}, \alpha_{li}^{L}, \alpha_{ki}^{K} \ge 0 \qquad \forall k,l,i \tag{3.46}$$

Constraints (3.45) and (3.46) are domain constraints.

Capacity constraints

$$S_j^{klt} \ge \sum_i X_{ji}^{klt} - \sum_i X_{ij}^{kl(t-e_{ij})} \qquad \forall k,l,t,j; t - e_{ij} \ge 1 \tag{3.47}$$

Constraint (3.47) keeps track of the stations where the driver picks the parcel that leave for transfer

$$r_j^t = r_j^{t-1} + \sum_k \sum_i Y_{ij}^{k(t-e_{ij})} - \sum_k \sum_j Y_{ji}^{kt} \qquad \forall j,t,i \ne j; t - e_{ij} \ge 1 \tag{3.48}$$

$$r_i^t \le R_i \qquad \forall i,t \tag{3.49}$$

Constraints (3.48) and (3.49) are storage capacity constraints for each locker at each time period.

$$\sum_{k} Y_{ij}^{kt} \leq V_l \qquad \forall l, i, j, t \tag{3.50}$$

Constraint (3.50) comprises the capacity constraints for the drivers at each time period.

Chapter 4 Solution Algorithms

Through the use of mathematical models, it is possible to obtain exact solutions. However, to obtain an optimal solution this way is too time consuming, especially when the problem is large. Therefore, we hope to develop efficient heuristic approaches to approximate an optimal solution within a shorter time than would be the case using a mathematical model. In this chapter, we propose two algorithms to solve this problem.

4.1 A Proposed Greedy Algorithm based on Direct Shipping

In this algorithm, we reduce the problem to consider only direct shipping, where the additional distance is not limited, and each driver can serve only one request at a time, which means that the driver must deliver the request to its destination before accepting another request. The idea of this algorithm is to try to transform the original network and use it to solve the matching problem by solving the minimum-cost flow problem.

Table 4.1 shows the steps for the greedy algorithm on a directed network $G=(N,A)$, where N is the set of nodes, and A is the set of arcs that represents the road network. A distance d_{ij} and travel time e_{ij} are associated with each arc $(i,j)\in A$. Also, L_{list} is a list of the drivers associated with an origin O_l^L to a destination W_l^L with the earliest time E_l^L and latest time G_l^L. K_{list} is a list of the parcels associated with an origin O_k^K to a destination W_k^K with the earliest time E_k^K and latest time G_k^K. P_{list} is the parcel list that has not been shipped by driver l.

For initialization the $P_{list}=L_{list}$. In step 1, we check the list of the parcels P_{list} that have to be delivered. Then, in step 2, we compute the $AD_{k,l}$, which is the additional distance for driver l in L_{list} if he/she delivers parcel k in P_{list}, where s_l is the shortest distance for driver l from the origin O_l^L to the destination W_l^L; s_k is shortest distance for the parcel k from its origin O_k^K to its destination W_k^K; $B_{k,l}$ is the shortest distance from the driver's origin to the parcel's origin, and $E_{k,l}$ is shortest distance from the parcel's destination to the driver's destination. Thus, the additional distance can be calculated using $AD_{k,l}=(B_{k.l}+s_k+E_{k,l})-s_l$. Note that all the shortest distances are calculated using Dijkstra's algorithm.

Table 4.1 Greedy Algorithm

Greedy Algorithm	
Data: $G, L_{list}, O_l^L, W_l^L, E_l^L, G_l^L, K_{list}, O_k^K, W_k^K, E_k^K, G_k^K$	
Initialization: $P_{list} = L_{list}$	
1: **While** $P_{list} \neq \varnothing$ **do**	
2:	Compute $AD_{k,l}$ for each parcel k in K_{list} for each driver l in L_{list}
3:	Compute $FT_{k,l}$ and $TC_{k,l}$ for each parcel k in K_{list} for each driver l in L_{list}
4:	Solve Min-Cost Flow problem
5:	Update P_{list}, O_l^L, E_l^L
6: **End While**	
Result: S	

After that, in step 3, we compute the $FT_{k,l}$, which is the finish time for driver l if he/she delivers parcel k, as calculated using E_l^L, the earliest departure time of driver l or E_k^K, the earliest departure time of parcel k, depending on which one comes later, plus the travel time if driver l delivers parcel k, which is calculated based on the additional distance $AD_{k,l}$. $TC_{k,l}$, which is the cost if the driver delivers the parcel, calculated from the $AD_{k,l}$ and the penalty cost for driver l who delivers parcel k, where the finish time $FT_{k,l}$ violates the driver's latest arrival time G_l^L or the latest arrival time G_k^K for parcel k.

After calculating the finish time $FT_{k,l}$ and cost $TC_{k,l}$ in steps 2 and 3, the problem can be transformed into a minimum-cost flow problem. The network includes a dummy driver node D_l to deal with excess parcels when the number of parcels P is more than the number of drivers L that have an arc capacity $\max\{K,P\}-K$ and a dummy parcel node D_k to deal with excess drivers when the number of parcels P is less than the number of drivers L that have an arc capacity $\max\{K,P\}-P$. We set supplies at the source node equal to $\max\{K,P\}$ and have supplies at the sink node equal to $-\max\{K,L\}$. The arcs connecting the source node to all drivers have costs equal to 0 with a capacity equal to 1 except for driver D_l.

Similarly, the arcs connecting from all of the parcels to the sink node have costs equal to 0 with a capacity equal to 1 except parcel D_k. All drivers except driver D_l will have arcs connecting to all of the parcels with cost $TC_{k,l}$ except the arc to the parcel D_k whose cost is M. The arcs from driver D_l to all of the parcels also have a significant cost M. Figure 4.1 illustrates our minimum cost flow network example.

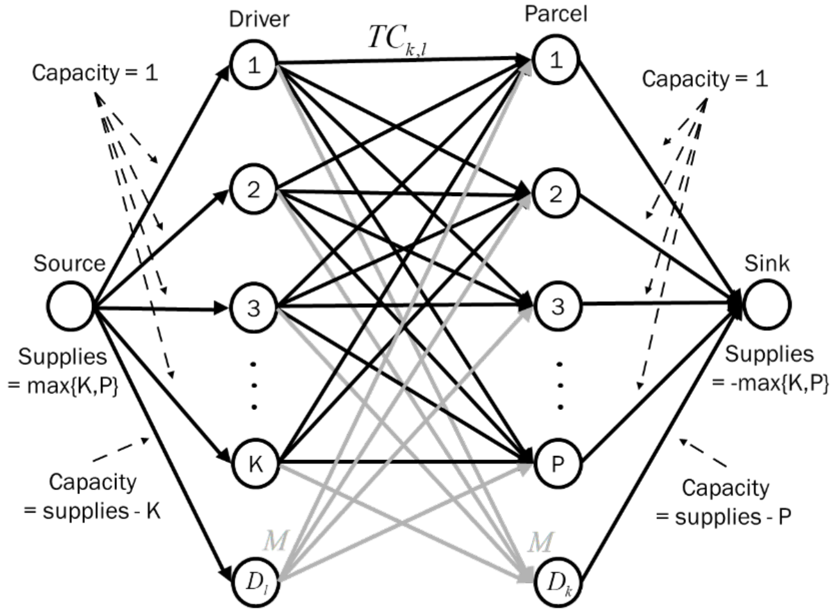

Figure 4.1 The minimum-cost flow schematic network

To solve the minimum-cost flow problem in step 4, we can use the following linear programming (LP) formulation as follows:

Parameters

$b(i)$	Supply at each node $i \in N$
$c_{i,j}$	Arc cost if passing arc $(i, j) \in \varepsilon$
$u_{i,j}$	Capacity of arc $(i, j) \in \varepsilon$

Decision Variable

$x_{i,j}$	Binary Variable equal to 1 if one chooses to pass through arc $(i, j) \in \varepsilon$; and 0 otherwise

The objective is to minimize the total cost for shipping all the parcels, as shown in the constraint (4.1)

$$\min \sum_{(i,j)\in\varepsilon} c_{i,j} x_{i,j} \tag{4.1}$$

Subject to

$$\sum_{i,j\in\varepsilon} x_{i,j} - \sum_{i,j\in\varepsilon} x_{j,i} = b(i) \qquad \forall i \in N \tag{4.2}$$

Constraint (4.2) is mass balance constraint

$$0 \le x_{i,j} \le u_{i,j} \qquad \forall (i,j) \in \varepsilon \tag{4.3}$$

Constraint (4.3) is flow bound constraint

After obtaining the optimal solution from step 4, we now know which parcel k will be delivered by which driver l and when it will arrive at its destination. Then, in line 5, we need to update the parcel list P_{list} that is the list containing parcel k that has not been sent by any driver (delivered by dummy driver D_l), and we must also update driver origin O_l^L and the earliest departure time E_l^L , where if the driver delivers the parcel, the driver's origin O_l^L will be the destination of parcel W_k^K , and the arrival time at the parcel's destination will be earliest departure time for driver E_l^L . After updating the information, steps 2-4 are repeated until all the parcels have been delivered by the driver ($P_{list} = \varnothing$).

4.2 A Proposed Rolling Horizon Algorithm

In this algorithm, we attempt to split the original problem into subproblems by splitting the total time into multiple time periods. In each time period, we consider only the driver and parcel that appear in that time period, which reduces the size of the problem. Then, we use our mathematical model to solve them sequentially. Figure 4.2 shows the rolling horizon framework.

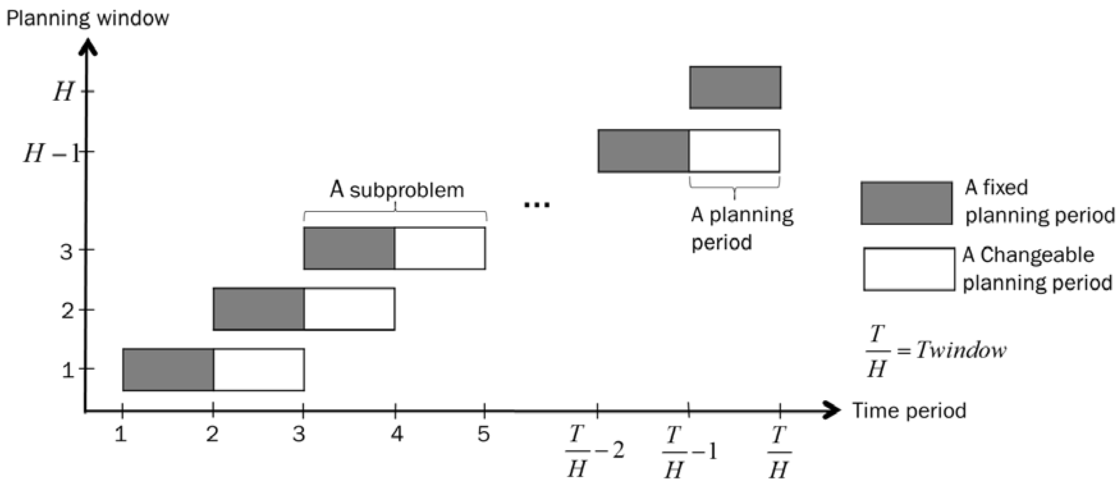

Figure 4.2 The proposed rolling horizon algorithm framework

Table 4.2 shown the steps for the rolling horizon algorithm used to select the driver and parcel in each time period. It requires information related to the earliest departure time E_k^K, E_l^L and latest arrival time G_k^K, G_l^L for both parcels in the K_{list} and drivers in the L_{list}. Also, the total time T for the original problem and the number of planning windows H are used to split the problem into subproblems. An initialization procedure is conducted before starting the algorithm $P_{list} = L_{list}$, $D_{list} = K_{list}$, $Twindow$ for a limited time for each time period, which equals $Twindow = T / H$, where h is the number of planning periods for each subproblem.

To build the subproblem for each time period, it is necessary to select the driver and parcel added to the subproblem. The driver and parcel were selected by checking the earliest departure time for each parcel and each driver. If it was not more than time period (i) minus one plus the number of planning periods (h) multiplied by $Twindow$, then we added that parcel and driver into the subproblem, as shown in lines 2 to 11. After that, we solved the subproblem using our mathematical model described in Chapter 3. After obtaining the result from the subproblem, we saved the result if that result or the latest arrival time for the driver and parcel was less than that time period (i) multiplied by $Twindow$, and we updated P_{list} and D_{list} by removing the parcel and the driver that we already saved, as shown in lines 13 to 22. Then, we repeated the algorithm from lines 2 to 22 again until the planning window H was completed.

Table 4.2 Rolling Horizon Algorithm

Rolling Horizon Algorithm
Data: $L_{list}, E_l^L, G_l^L, K_{list}, E_k^K, G_k^K, T, H, h$
Initialization: $P_{list} = L_{list}, D_{list} = K_{list}, Twindow = T / H$
for $i = 1\ to\ H$ **do**
for $j\ in\ D_{list}$ **do**
if $E_j^L \le (i + h - 1) \cdot Twindow$ **then**
add driver j to subproblem
End if
End for
for $k\ in\ P_{list}$ **do**
if $E_k^K \le (i + h - 1) \cdot Twindow$ **then**
add parcel k to subproblem
End if
End for
Solve subproblem
for $j\ in\ D_{list}$ **do**
if driver *j* is scheduled or $G_j^L \le i \cdot Twindow$ **then**
save driver *j* schedule and update D_{list}
End if
End for
for $k\ in\ P_{list}$ **do**
if parcel k is scheduled or $G_k^K \le i \cdot Twindow$ **then**
save parcel k schedule and update P_{list}
End if
End for
End for
Result: S

4.3 Summary

In this chapter, we introduced two algorithms. The first was a greedy algorithm used to solve our problem while considering only direct shipping, where the driver could only ship the parcels one at a time. The second algorithm (the rolling horizon algorithm) was used to split the original problem into multi subproblems.

Chapter 5 Computational Experiments

In this chapter, we describe how we generate the experimental settings for our problem and the computational results for our model and algorithm. Our testing was implemented in Python and conducted on an Intel Core i7-4770 CPU @ 3.40GHz*8, 16 GB RAM computer. Gurobi version 8.1.1 was used to solve the integer programs.

5.1 Experimental settings

The networks that we used to conduct the experiment were in the form of a grid network. The number of arcs was twice the number of nodes, and we tested 4 settings on the number of nodes: 15, 25, 50, and 100. For each test network, the number of locker nodes was 80% of the total number of nodes. The weight of each arc was randomly set between 1 to 3. To generate all the driver and parcel information first, we randomly chose the origin and destination for all of the drivers from the graph, and all the parcel origins and destinations were randomly chosen from the locker node in the graph. After that, for all the drivers and parcels, we found the shortest distance from their origins to their destinations. Last, we randomly generated the earliest departure time between 1 and 48 (representing the operation time 4 hour increments since each period was 15 min, and we assumed that the operational time started at 8am and ended at 8pm). We then randomly generated the latest arrival time, which a had a time limit between their earliest departure time plus their shortest travelling time and a minimum between 48 and summation of their earliest departure time and three times their shortest travelling time. The cost- and capacity -related parameters are summarized in Table 5.1.

Table 5.1 Cost and capacity related parameters

Parameter	Value
V_l	5
c_k	$100+(4\times s_k)$
w_1	2/km
w_2	8
w_3	10/hr
c_p	1000

5.2 Experimental results

In this section, we report the experimental results as divided into the three subsections. Subsection 5.3.1 shows the experimental computation results for the proposed mathematical models. Subsection 5.3.2 shows the experimental computation results for the proposed algorithm. Subsection 5.3.3 shows the experimental computation results when combining the two algorithms.

All scenarios limited the running time to 7200 seconds because no significant improvement would occur by increasing this time limit. If the optimal solution couldn't be derived within the time limit, the solution would be the most feasible solution obtained in 7200 seconds.

5.2.1 Mathematical models result

Initially, we compared the result of our model using the locker as a pickup, drop off, and transfer location, based on Chen et al. (2017) and by using total cost as an indicator, for which the results are shown in Table 5.2. The input parameters are the same for all models, which make the storage capacity of the locker more than enough. M^A refers to the arc-based model, and M_T^N refers to the time-space network model for the case with 15 and 25 nodes, which had 30 and 50 arcs, respectively. Here *n* and *m* represent the number of nodes and number of arcs for the testing network, respectively. *L* represents the number of drivers; *K* represents the number of parcels. *T* represents the index of the last period for the M_T^N model. CPU Time represents the total computational time necessary to obtain the solution. *Save* represents the percentage that our model can save in terms of total cost compared to Chen et al.'s model.

According to the results shown in Table 5.2, our model can save more cost for all cases due to the time synchronization constraints, where the drivers do not need to wait for another driver to transfer the parcel; they can just put the parcel into the locker. If the driver has only enough time and must detour to carry a parcel to put it at an intermediate node but does not have enough time to wait for next driver, our model can reduce the cost for that case. Model M^A can save around, total cost while model M_T^N in a small case can save up to 10-40% of the total cost, but for a medium case, it can only save around 3-15% because it cannot reach the optimal solution within the7200 second time limit.

Table 5.2 Model comparison

n	*m*	*L*	*K*	Without locker	M^A		M_T^N	
				CPU Time (s)	***Save* (%)**	**CPU Time (s)**	***Save* (%)**	**CPU Time (s)**
15	30	5	10	115	10.21	125	10.21	242.57
		10	5	577.06	38.72	356.45	38.72	519.15
		10	10	506	22.06	367.5	22.06	560.65
25	50	10	15	7200	8.42	7200	8.42	7200
		15	10	7200	30.44	7200	15.72	7200
		15	15	7200	13.48	7200	10.95	7200

The performance of our models is shown in Table 5.3. The *n* that we use for testing our model included 15, 30 and 60, for which the results reported here are averaged over 5 runs for each problem instance. *Gap* is representing the gap between the best solution and the best bound obtained from the Gurobi software. *T* represents the index of the last period for the M_T^N model.

Table 5.3 Performance summaries

n	*m*	*L*	*K*	M^A		M_T^N		
				Gap (%)	**CPU Time (s)**	***T***	***Gap* (%)**	**CPU Time (s)**
15	30	5	10	0	125	48	0	242.57
		10	5	0	356.45	48	0	519.15
		10	10	0	367.5	48	0	560.65
25	50	10	15	16.35	7200	48	60.5	7200
		15	10	69.39	7200	48	88.2	7200
		15	15	66.01	7200	48	94.7	7200
50	100	30	15	-	7200	48	-	7200
		50	25	-	7200	48	-	7200
		100	50	-	7200	48	-	7200

Note: - means the model cannot find any feasible solution within time limit 7200 (s)

According to the results shown in Table 5.3, L is one of the important factors that affects the model performance. A larger L value requires a longer solution time because the model has to find the path for all of the drivers (L) even when the drivers don't engage with the parcels (K). However, when L is greater, this means there are more opportunities to match the parcel and the driver.

5.2.2 Algorithm results

We attempted to reduces the computational time by using the solution obtained from the greedy algorithm as an initial solution. Thus, we used 15, 30 and 60 cases as a comparison. The results when comparing among only M^A, M_T^N and M^A, M_T^N using the initial solution from the greedy algorithm results are show in Table 5.4. The computational time for the greedy algorithm is shown in the brackets.

Table 5.4 Computational results for M^A and M_T^N with and without greedy algorithm

					M^A				M_T^N			
					Without Greedy		**with Greedy**		**Without Greedy**		**with Greedy**	
n	***m***	***T***	***L***	***K***	**CPU Time (s)**	**Gap (%)**	**CPU Time (s)**	**Gap (%)**	**CPU Time (s)**	**Gap (%)**	**CPU Time (s)**	**Gap (%)**
15	30	48	5	10	125	0	66.47 (0.47)	0	242.57	0	124.78 (0.47)	0
			10	5	356.45	0	147.81 (0.78)	0	519.15	0	245.91 (0.78)	0
			10	10	367.5	0	150.28 (1.16)	0	560.65	0	261.66 (1.16)	0
25	50	48	10	15	7200	16.35	3802.23 (0.98)	0	7200	60.5	7200 (0.98)	13.91
			15	10	7200	69.39	4851.5 (1.91)	0	7200	88.2	7200 (1.91)	35.85
			15	15	7200	66.01	5216.42 (2.63)	0	7200	94.7	7200 (2.63)	28.42
50	100	48	15	30	7200	102.51	7200 (16.26)	36.52	7200	-	7200 (16.26)	-
			30	15	7200	122.84	7200 (15.96)	68.96	7200	-	7200 (15.96)	-
			30	30	7200	107.9	7200 (17.09)	72.7	7200	-	7200 (17.09)	-

Note: - means the model cannot find any feasible solution within time limit 7200 (s)

As shown in Table 5.4, by using the solution from the greedy algorithm as an initial solution to solve the mathematical model, the mathematical model can get the optimal solution faster compared to not using the initial solution from the greedy algorithm. However, it's still cannot deal with M_T^N for the large network size of our testing.

We also tested our rolling horizon algorithm with different network sizes, including 15, 30, and 60 cases; *H* represents the number of the time periods that will be split from the total time into subproblems. *B-Gap* represents the gap found when comparing the results between the original model and the rolling horizon algorithm.

$$B - Gap = \frac{Obj.IP - Obj.Rolling}{Obj.IP} \times 100$$

Table 5.5 shows the computational result for M^A and M_T^N compared with the rolling horizon algorithm.

Table 5.5 Computational result for the rolling horizon algorithm

n	*H*	*T*	*L*	*K*	M^A			M_T^N		
					IP	**Rolling Horizon**		**IP**	**Rolling Horizon**	
					CPU Time (s)	**CPU Time (s)**	***B-Gap*** **(%)**	**CPU Time (s)**	**CPU Time (s)**	***B-Gap*** **(%)**
15	6	48	5	10	125	8.38	0	242.57	15.95	0
			10	5	356.45	18.89	0	519.15	49.14	0
			10	10	367.5	28.44	0	560.65	64.25	0
25	6	48	10	15	7200	1313.69	(10.35)	7200	1627.42	(19.53)
			15	10	7200	1827.27	(49.93)	7200	2488.47	(34.82)
			15	15	7200	1914.76	(41.87)	7200	2612.50	(38.12)
50	6	48	15	30	7200	4893.26	(36.54)	7200	5794.98	-
			30	15	7200	5728.53	(52.48)	7200	6477.31	-
			30	30	7200	6126.54	(60.46)	7200	6986.00	-

Note: - means the model cannot find any feasible solution within time limit 7200 (s)

The results shown in Table 5.5 indicate that using the rolling horizon algorithm to split the problem into multiple subproblems makes it possible for the model to deal with a large problem and to obtain a feasible solution within the time limit, but this may not be the optimal solution for a large problem. However, in the case of a small size problem, the solution obtained from the algorithm is the optimal solution, while for middle and large problems, a better solution can be obtained within a shorter time. Using M_T^N to solve large cases, we cannot compare the results for the large problem since the original model cannot get a feasible solution within the time limit.

5.2.3 Other experimental results

The results from Section 5.2.2 suggest that the proposed greedy algorithm does speed up the mathematical models, and the rolling horizon algorithm may be useful in practice because it can use the most updated information as a fixed initial input to plan for the next few periods.

In this section, we combine the two algorithms and increase the number of cases to 25, 50, and 100, where *K* is 25, 40, and 50 when *L* is twice *K*, and *T* is reduced from an entire day (48 time periods) to 2 hours (8 time periods). We then compare the computational results between M^A, M_T^N with and without an initial solution from the greedy algorithm. Table 5.6 provides the computational results for M^A with and without an initial solution with *T* limited to 8 time periods.

Table 5.6 Computational results for M^A with and without an initial solution (limit T = 8)

n	***m***	***T***	***L***	***K***	M^A		M^A **with initial**	
					GAP (%)	**CPU Time (s)**	**GAP (%)**	**CPU Time (s)**
25	50	8	50	25	0	4251.14	0	2517.62
			80	40	0	5189.58	0	3070.35
			100	50	0	6947.17	0	4270.48
50	100	8	50	25	0	5815.42	0	3615.55
.			80	40	85.74	7200	0	6285.28
			100	50	90.14	7200	40.14	7200
100	200	8	50	25	-	7200	35.15	7200
			80	40	-	7200	43.57	7200
			100	50	-	7200	70.18	7200

Note: - means the model cannot find any feasible solution within time limit 7200 (s)

As shown in Table 5.6, M^A using an initial solution for the small network model leads to the optimal solution faster by around 40%. The medium network can also obtain the optimal solution except for the case that has 100 drivers and 50 parcels, where the optimal solution cannot be derived, but a feasible solution with a smaller GAP can be derived. The largest network M^A with an initial solution cannot obtain the optimal solution for all cases, while that without the initial solution cannot get any feasible solution within the time limit of 2 hours.

Table 5.7 shows the computational results for M_T^N with and without an initial solution with a time limit *T* of 8 time periods.

Table 5.7 Computational results for M_T^N with and without initial solution (limit *T* = 8)

n	***m***	***T***	***L***	***K***	M_T^N		M_T^N **with initial**	
					GAP (%)	**CPU Time (s)**	**GAP (%)**	**CPU Time (s)**
25	50	8	50	25	0	1247.21	0	635.89
			80	40	0	2470.53	0	1297.67
			100	50	0	4847.84	0	2602.89
50	100	8	50	25	0	2751.56	0	1484.23
			80	40	0	4975.84	0	2510.25
			100	50	-	7200	50.53	7200
100	200	8	50	25	45.15	7200	0	4836.34
			80	40	-	7200	58.47	7200
			100	50	-	7200	83.54	7200

Note: - means the model cannot find any feasible solution within time limit 7200 (s)

The results shown in Table 5.7 indicate that M_T^N using an initial solution for the small and medium networks can get the optimal solution faster by almost 50% for most of the cases, except for the medium network with 100 drivers and 50 parcels, for which the optimal solution cannot be obtained, but some feasible solutions can be derived. The largest network M_T^N with an initial solution can get the optimal solution only for the case that has 50 drivers and 25 parcels. However, for the other cases, the optimal solution cannot be derived, but some feasible solutions can be obtained within time limit of 2 hours.

The results from Table 5.6 and Table 5.7 using an initial solution and time limit T can obtain the optimal solution faster for the small and medium network M_T^N than M^A, except for the medium network with 100 drivers and 50 parcels, where M^A can get a feasible solution with a smaller GAP. The largest network M_T^N can obtain the optimal solution only for the case with 50 drivers and 25 parcels. For the other cases, M_T^N can obtain some feasible solutions within the time limit but the GAP is larger than that of M^A.

5.3 Summary

In this chapter, we discuss our experimental results. Initially, we compared both of our models in which a locker was used as the pickup, drop off, and transfer location with the model where the locker was not used. The results show that both of our models saved up to 40% of the total cost. When the number of the drivers was more than the number of parcels, this saved even more, but it was more time consuming. The computational experiments showed that our models cannot deal with large cases. Therefore, we tried to reduce the computational time of our mathematical model by developing two algorithms. The results showed that the use of a greedy algorithm and a rolling horizon algorithm can make the mathematical model solve faster, but it still cannot deal with a large case. Lastly, we combined the two algorithms by reducing the time from an entire day to 2 hours. The results showed that reducing the time, in general, M_T^N could obtain the optimal result faster for the small and medium networks. For the large network, M^A and M_T^N could obtain some feasible solutions, and if there was a small number of drivers and parcels , M_T^N could obtain the optimal result.

Chapter 6 Conclusion and Future research

6.1 Conclusion

In this research, we considered a distribution network that is mostly operated by private drivers and has professional drivers as a backup option, where a smart locker is used as a pickup, drop off, and transfer location for all the parcels. The crowdsource delivery platform takes advantage of the spare capacity in the private vehicles of commuters along with their scheduled trips. The use of smart lockers was targeted on unsuccessful deliveries and consolidation opportunities. Certainly, locker pickup is a convenient pickup option for consumers, while potentially driving delivery costs down by reducing the number of delivery points and avoiding unsuccessful deliveries that lead to multiple delivery attempts. In order to integrate these two solutions by using crowdsourcing to ship parcels via a network of lockers, our problem was how to plan trips for the individuals delivering the parcels, where all drivers and parcels have their own origin, destination, earliest departure time, and latest arrival time. Our research is a multi-hop rider matching problem, where the time window is emphasized. We conclude this research and summarize our contributions as below:

1. This research may be arguably the first (among problem setting and mathematical models) using smart lockers as transhipment nodes, which gives better utilization of the spare time of drivers and the use of smart lockers.
2. Transhipment via smart lockers should be better than via drivers since the latter requires synchronization between contiguous drivers to carry out multi-hop shipping.
3. We proposed an arc-based model modified from the literature. This model is more suitable for larger capacity lockers since it can at most limit the used capacity over the entire planning horizon (instead of at any period like the time-space model below) for one locker.
4. We proposed a time-space network model aimed at dealing with the locker capacity for any period. This model can correctly calculate detailed movement and keep track of locker capacity, but is more time consuming than the arc-based model discussed above.
5. We proposed a greedy algorithm that only considers direct shipping (i.e., no transhipment), and then used the solution as a warm start initial solution for both the

arc-based and time-space network models. This saved computational time for mathematical programming models.

6. We proposed a rolling horizon algorithm, which split the problem into multiple subproblems in a rolling-horizon fashion. Then, we sequentially solved each subproblem using our integer programming model.
7. We conducted some computational tests, whose results suggest our proposed greedy algorithm does speed up our mathematical models, and the rolling horizon algorithm may be useful in practice because it can use the most updated information as a fixed initial input to plan for the next few periods.
8. We conducted more computational tests in which the length of the time periods was considered. The results showed that using the initial solution can speed up our model by around 50%. Our time-space network can obtain the optimal result faster than the arc-based model in small and medium networks. For large networks, in most of the cases, our model using an initial solution could only obtain a feasible result within the required time limit.

6.2 Future research

Because smart lockers are a new innovation, previous research on their use is still limited. We thus encountered several difficult and challenging problems. We list several that we consider to be important for future researchers to investigate as follows:

1. Develop a better algorithm to deal with transhipment

Based on the high computational complexity of our model, our proposed greedy algorithm is still limited to solve only for direct shipping. Developing an algorithm that can deal with transhipment for crowdshipping is a challenging problem that could be investigated.

2. Locker station facility location problem

To choose a place to construct the locker station, if we can determine all the demand information using our model, we will also know which area has enough drivers to make the transshipments. Then, we can decide the location and storage capacity for each locker station, which can cover most of the demand also can be the transhipment point. This might improve the utilization of locker stations.

3. Locker sizes for different size parcels

In the real world, for each locker station, there may be different sizes and numbers of locker boxes. For example, station A might have 15 small size boxes, 10medium size boxes and 5 large size boxes, and station B may have 20 small size boxes and 15 medium size boxes, etc. Also, the size of the parcels typically differs, which affects the capacity of the driver, who also has to match the size of the parcel to the locker boxes. This problem will become more challenging and interesting if considering the different sizes of the locker boxes and the parcels.

Reference

Agatz, N., Erera, A., Savelsbergh, M., Wang, X. (2010). Sustainable passenger transportation: Dynamic ride-sharing.

Agatz, N., Erera, A., Savelsbergh, M., Wang, X. (2011). Dynamic ride-sharing: a simulation study in metro Atlanta. *Transportation Research Part B,* 45(9), 1450-1464.

Agatz, N., Erera, A., Savelsbergh, M., Wang, X. (2012). Optimization for dynamic ride-sharing: a review. *European Journal of Operational Research*, 223(2), 295-303.

Baldacci, R., Maniezzo, V., Mingozzi, A. (2004). An exact method for the carpooling problem based on lagrangean column generation. *Operations Research*, 52(3), 422-439.

Chen, W., Mes, M., Schutten, M. (2017). Multi-hop driver-parcel matching problem with time windows. *Flexible services and manufacturing journal,* 1-37.

Coltin, B., Veloso, M. (2014). Ridesharing with passenger transfers. In *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp 3278-3283.

Cordeau, J., Laporte, G. (2007). The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, 153 (1), 29-46.

Cortés, C., Matamala, M., Contardo, C. (2010). The pickup and delivery problem with transfers: formulation and a branch-and-cut solution method. *European Journal of Operational Research*, 200 (3) 711-724.

Deutsch, Y., Golany, B. (2017). A parcel locker network as a solution to the logistics last mile problem. *International Journal of Production Research,* 56:1-2, 251-261.

Faugere, L., Montreuil, B. (2016). Hyperconnected City Logistics: Smart Lockers Terminals and Last Mile Delivery Networks. In *Proceedings of 3rd International Physical Internet Conference*.

Faugere, L., Montreuil, B. (2017), Hyperconnected pickup and delivery locker networks. In *Proceedings of the 4th International Physical Internet Conference.*

Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.-E., Wang, X., Koenig, S. (2013). Ridesharing: the state-of-the-art and future directions. *Transportation Research Part B*, 57, 28-46.

Herbawi,W., Weber,M. (2011a). Evolutionary multiobjective route planning in dynamic multi-hop ridesharing. *European conference on evolutionary computation in* *combinatorial optimization*, pp 84-95.

Herbawi, W., Weber, M. (2011b). Comparison of multiobjective evolutionary algorithm for solving the multiobjective route planning in dynamic multi-hop ridesharing. In: *Proceedings of the 11th European Conference on Evolutionary Computation in Combinatorial Optimization*, pp 84-95.

Herbawi, W., Weber, M. (2012). A genetic and insertion heuristic algorithm for solving the dynamic ridematching problem with time windows. In *Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference*, pp 385-392.

Hosni, H., Naoum-Sawaya, J., Artail, H. (2014). The shared-taxi problem: formulation and solution methods. *Transportation Research Part B*, 70, 303-318.

Kämäräinen, V. (2001). The Reception Box Impact on Home Delivery Efficiency in the E-Grocery Business. *International Journal of Physical Distribution & Logistics Management*, 31 (6), 414-426.

Lemke, J., S. Iwan, and J. Korczak (2016). Usability of the Parcel Lockers from the Customer Perspective: The Research in Polish Cities. Transportation Research *Procedia*, 16: 272-287.

Liaw, C.-F., White, C.C., Bander, J. (1996). A decision support system for the bimodal dial-a-ride problem. *Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 26 (5), 552-565.

Masson, R., Lehuédé, F., Péton, O. (2014). The dial-a-ride problem with transfers. *Computers & Operations Research*, 41(0), 12-23.

Masson, R., Trentini, A., Lehuédé, F., Malhéné, N., Péton, O., Tlahig, H. (2015). Optimization of a city logistics transportation system with mixed passengers and goods. *EURO Journal on Transportation and Logistics*. 1-29.

Montreuil, B., R. D. Meller, and E. Ballot (2010). Towards a Physical Internet: The Impact on Logistics Facilities and Material Handling Systems Design and Innovation. *Progress in Material Handling Research*. 305-327.

Pitney Bowes Parcel Shipping Index forecasts 20 percent industry growth by 2018. (2018, October 16). Retrieved from https://www.pitneybowes.com/us/shipping-and-mailing /case -studies/pitney-bowes-2017-parcel-shipping-index.html

Punakivi, M., and Yrjölä H., Holmström J. (2001). Solving the Last Mile Issue: Reception Box or Delivery Box?. *International Journal of Physical Distribution & Logistics*

Management, 31 (6), 427-439.

Savelsbergh, M.W.P., Sol, M. (1995). The general pickup and delivery problem. *Transportation Science*, 29 (1), 17-29.

Stein, D.M. (1978). Scheduling Dial-a-Ride Transportation Systems. *Transportation Science* 12 (3), 232-249.

Stiglic, M., Agatz, N., Savelsbergh, M., Gradisar, M. (2015). The Benefits of Meeting Points in Ride-Sharing Systems. *Transportation Research Part B*, 82, 36-53.

Wolfler Calvo, R., de Luigi, F., Haastrup, P., Maniezzo, V. (2004). A distributed geographic information system for the daily carpooling problem. *Computers & Operations Research*, 31 (13), 2263-2278.