University of Padova

Master's Degree in ICT for Internet and Multimedia

Department of Information Engineering

Food Order Delivery using Crowdshipping: A MILP-based Rolling Horizon Approach

Supervisor
Dr. Roberto Roberti
University of Padova

Master Candidate
Vahnika Meka

Student ID
2051163

Academic Year
2023-2024

Abstract

In recent years, the advancement in technologies has led to creative solutions like crowdshipping which deals with the use of occasional drivers in the delivery systems to lower the cost of transportation, and a small compensation fee is paid to the occasional drivers for their service. This paper considers a food delivery system, which incorporates occasional drivers in addition to dedicated vehicles present at each restaurant and makes decisions about the assignment of orders while minimizing the total cost and delivering the order within its time window. Here, each occasional driver has a detour radius and order delivery can be done by a single occasional driver or by transfer between two occasional drivers respecting their detour radii or by a restaurant's dedicated vehicle. An event-based rolling horizon approach is proposed, at each decision epoch a mixed-integer linear programming model is used to solve the problem, and if the dedicated vehicles are used for the delivery, then a routing algorithm is implemented to build dedicated vehicles' routes. The model guarantees the delivery of orders to the customers and a penalty is imposed if the delivery time exceeds the order's time window. CPLEX solver is used to implement this model and computational analysis done on the instances demonstrates the impact of different configurations of restaurants, occasional drivers, and customers on key performance metrics such as total cost, order fulfillment rate, on-time arrival rate, and pending orders rate. The results show that this approach provides good solutions within short run times and transfer delivery by occasional drivers is a cost-effective way to reduce overall delivery cost.

Keywords: Crowdshipping, Occasional drivers, Rolling Horizon Approach, Mixed-Integer Linear Programming (MILP), Heuristics

Contents

Listing of figures

Listing of tables

Listing of acronyms

Mixed Integer Linear Programming . . . . MILP

Occasional Driver . . . . OD

Dedicated Vehicle . . . . DV

1 Introduction

The way the world eats has changed drastically over past few years. A couple of decades ago, food delivered by restaurants was mainly limited to certain food categories like pizzas and Chinese. But there has been a tremendous increase in online food delivery services due to the COVID-19 pandemic and later it continued since people started to prefer to order most things online at the ease and comfort of their homes. In the recent survey, it is shown that 70% of customers prefer ordering food online and 37% believe that ordering via a food delivery app is the most reliable which shows how much people love the convenience of online ordering [1]. It is crucial to satisfy such huge demand, and the key challenge is the logistics to deliver the meals from the restaurant to customers. The delivery companies generally offer an app or website listing the restaurants and their menu. The customer selects their order, the total amount to be paid is displayed and when the customer places an order, the estimated arrival of the order is displayed. The global revenue of the online food delivery market is forecasted to be 1.22 trillion US dollars in 2024, which exhibits a compound annual growth rate of 10.06% [2].

There are several stakeholders in this process of delivering food orders, customers needing fast and quality services, restaurants wanting more customer engagement, and then the drivers to serve the customers. The delivery platform will be successful if it satisfies these stakeholders and profits when more customers order using their app or website. The goodwill from customers will increase when they receive their food fresh and quickly. According to McKinsey's article, the critical factor in customer satisfaction is the speed of delivery without compromising the quality [3]. The report mentions that this trend is driven by growing reliance on online platforms for food delivery, where consumers prioritize convenience and speed. As the food delivery industry continues to evolve, crowdshipping emerges as a dynamic and flexible model, harnessing the power of everyday travelers to meet the growing demand for quick and cost-effective meal delivery.

The term crowdshipping refers to the concept of involving a group of willing participants in a personalized delivery of goods i.e., an external agent volunteers to take part in the delivery of a product in return for a reward. In the early 2000s, this concept is applied to a few online ventures, where ordinary car drivers register as couriers to deliver parcels and get paid for the delivery. This helps the driver to earn an extra income, and the company gets the resources to deliver their product. The research paper in collaboration between CIRREALT Interuniversity Research Center, Université Laval, and Canada Research Chair in Interconnected Business Engineering states that crowdshipping delivery is an answer to the growing expectations of customers for faster, more personalized, and cost-efficient delivery service [4]. It encourages the passengers to use the extra space in their vehicle to carry other people's parcels. The only concern that is raised about the crowdshipping is the liability and security. Therefore, to acknowledge the concerns, the external agents must register and clear all the security verifications before they can deliver a product. These external agents are referred to as Occasional Drivers (ODs) and like any other job, a fee is paid to them for their service.

The integration of the crowdshipping concept into traditional food delivery services helps to meet growing consumer demands for speed, convenience, and sustainability. By combining crowdshipping with a restaurant's delivery fleet, food delivery platforms can optimize logistics, reduce costs, and enhance customer satisfaction. Crowdshipping leverages the travel routes of everyday people, who can pick up and deliver orders along their existing paths. When linked with a restaurant's vehicle fleet, this hybrid model provides several benefits. Restaurants can rely on their fleet for larger, more complex deliveries while using crowdshipping to handle smaller, more frequent orders that require quick turnarounds. This not only maximizes the efficiency of delivery routes but also ensures that customers receive their meals fresh and on time, even during peak hours.

In crowdshipping delivery, the occasional drivers are usually freelancers willing to transport the packages from the source to their destination. A few factors are limiting the delivery by occasional drivers i.e., the number of orders, and the locations of customers that one can handle depends on that occasional driver and the provider that assigns the orders. Therefore, it is beneficial to have a regular set of drivers, who reside at the supplier and they can handle any order regardless of the quantity and location of the customers. These standby vehicles are often referred to as Dedicated Vehicles (DVs), and their scheduling and maintenance are under the responsibility of the company or the source that dispatches the orders. The route of a dedicated vehicle is known as the path or sequence of orders that it must deliver and the dedicated vehicle starts its route from the company, delivers the orders, and returns to the company. Thus, this delivery by a dedicated vehicle causes a certain travel cost which is borne by the company and the travel cost includes the distance traveled by the DV, payment to the driver of the DV, and other costs such as gas and maintenance.

The food delivery system using crowdshipping must have a trade-off between the occasional drivers and dedicated vehicles during the assignment of the orders. The usage of occasional drivers is beneficial in terms of cost since compensation is paid for their services which is comparatively lesser than the cost incurred by a dedicated vehicle. It is important to consider the behavior and preferences of the occasional drivers who participate in these platforms. Many of the occasional drivers prefer to deliver orders that are conveniently along their route, particularly on their way home, rather than deviating significantly from their intended path. For instance, during peak hours or in areas that are harder to reach by the restaurant's fleet, crowdshipping can efficiently handle orders that align with a driver's existing route. This not only ensures quick delivery but also maximizes the number of deliveries that can be made without additional travel, making the process more efficient and sustainable. On the other hand, for deliveries that require significant detours, the restaurant's vehicles can be utilized to maintain control over delivery times and ensure customer satisfaction.

In this paper, the focus is on the delivery platform that optimizes the assignment of incoming food orders from customers to available delivery resources i.e., either to occasional drivers or to the regular dedicated vehicles or a pair of occasional drivers for a transfer delivery scenario. The platform assigns the orders to backup vehicles (i.e., dedicated vehicles) only when it is not feasible or efficient to use an occasional driver for the delivery of that order and determines the specific delivery route. The delivery of an order must happen within their time windows since most real-time restaurants work with food orders having time constraints. The timeline of a food order starts with the customer deciding to order food from a restaurant via a delivery app,

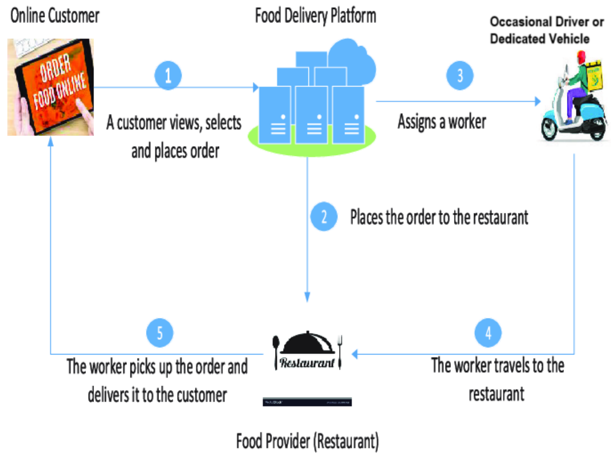

Figure 1.1: Process of Food Order Delivery

the app lists the pricing and estimated delivery time depending on the distance of the customer from the restaurant. Then the restaurant needs time to prepare the order, once it is ready the order can be delivered by the best available resource to the customer within the estimated delivery time, otherwise a penalty is imposed for the late delivery. This type of service is most relevant in same-day delivery settings such as online food delivery, grocery delivery, and delivery of medicines. The figure 1.1 depicts the entire timeline of the food order delivery.

The main contributions of this paper are as follows. Firstly, a dynamic food delivery model is designed that uses the concept of crowdshipping. It involves occasional drivers along with the restaurant's dedicated vehicle and an event based rolling horizon approach is developed to repeatedly solve the problem with the current information. This model introduces a new concept of transfer delivery by occasional drivers, where a food order is assigned to a pair of occasional drivers and a routing heuristic is proposed to plan routes for dedicated vehicle delivery. One occasional driver receives the order from the restaurant and then transfers it to another occasional driver to deliver the order to the customer. A computational study is performed on different instances that depict the real-time delivery scenarios to understand the capability of the proposed model.

The remainder of this research work is organized as follows: Section 2 is the literature review which discusses the relevant articles and studies published on the concept of crowdshipping, crowdsourcing, and occasional drivers. Section 3 presents the problem description which describes the model used in a mathematical form. Section 4 elaborates the method used to find the solution of the model. Section 5 shows the results of the computational experiments performed on instances of the proposed model and an explanation of the research findings. Finally, Section 6 provides the concluding remarks and scope of future research on this field.

2
Literature Review

This section summarizes the previous research on crowdsourcing and crowdshipping, compares the findings obtained during the analysis of articles, and finally identifies the research direction of our work. The articles that have been studied are found using a keyword in the database of Scopus. After screening them, seventeen papers are chosen that fall under the category crowdshipping and vehicle routing problem to perform the delivery of goods to reduce the transportation cost. The following table 2.1 outlines the literature work done in this research.

2.1 Assignment and Routing of the Vehicles in Deterministic Settings

Here, the problem considers deterministic settings where the events that occur over the period are known, and no randomness is associated with them. Since we deal with the assignment of orders to vehicles and route their trips in this environment, we have complete knowledge of the incoming orders from customers and the availability of vehicles along with their characteristics like the amount of demand and supply we can expect. The following articles summarize different research done in this setting while including different parameters in their applications.

Archetti et al. (2016) work introduces a deterministic Vehicle Routing Problem (VRP) with the company's fleet of vehicles are capacitated vehicles and the use of Occasional Drivers (OD) to perform delivery of goods. The author developed a model that includes compensation paid to ODs and its influence on the willingness of the OD to deviate from their route to do the delivery. The goal is to minimize the cost of transportation of regular vehicles along with the compensation paid to the OD and the solution is to assign routes to regular vehicles and OD that follow the objective. Here, the approach used to obtain the solution is a multi-start heuristic in combination with neighborhood search and tabu search. The findings from this article explain substantial cost savings by using Occasional Drivers if they are significant in number and have generous flexibility to make deliveries[5].

Later the work is extended in the article by Dahle et al. (2019) that considers muti-point pick-up and delivery locations with time windows and it follows the same objective. In this article, the drivers (both regular and occasional) can pick up the goods from different locations and deliver to different customers' locations and the delivery must be done within the maximum time limit assigned to each order in a deterministic setting i.e., all the available orders and the all drivers are known. The compensation fee paid to OD affects their willingness to perform delivery and four compensation schemes are proposed in this article following that the OD can make multiple pick-ups and deliveries, and the results show that the schemes attract the OD to participate in the delivery if they are not too low and creating a complex compensation scheme that mimics behavior of OD helps to yield more profits[6].

Zhen et al. (2022) study a logistic service platform that deals with internal company deliveries and serves external business deliveries within its delivery capacity. The pick-up of the internal and external orders is performed by either dedicated vehicles on dedicated routes that leave and return to the depot or occasional vehicles that have their destination location. The aim of this platform includes minimizing the total cost of routes by vehicles minus revenue generated by serving external orders. The problem is named a closed-open mixed multi-depot VRP and modeled as a mixed integer programming model with column generated-based heuristic (CG). This article proposes a CG-based solution method to provide cost-reduction benefits. It considers a real-world problem of logistics platform with large-scale instances including multi-depot, internal customers, external tasks, and two types of vehicles[7].

The work by Parappathodi and Archetti (2022), considers an application named Crowdsourced Humanitarian Relief VRP that uses crowdsourcing of volunteer drivers to deliver emergency kits during natural disasters. The settings are deterministic i.e., the availability of volunteer drivers and demand for relief kits is known. The article considers a multi-pickup depot to minimize time to serve the demand points since time is crucial in such a situation. The solution is obtained using iterated local search (ILS) heuristics to provide a quick and good match between the demand points with available volunteers to deliver the goods[8].

2.2 Assignment and Routing of the Vehicles in Dynamic and Stochastic Settings

In real-world applications, there is some randomness involved in the customer placing an order, the driver's willingness to take up a delivery, or the demand and supply to expect. The dynamic nature of the problem is to give better solutions over time by including the new information available at regular time intervals. The stochastic approach incorporates a priori knowledge of the uncertainties in either the orders, drivers, or both and produces solutions according to the fluctuations in them. The articles mentioned below include such dynamic and stochastic nature in their application and provide solutions closely related to real-time problems.

The article by Alp M. Arslan et al. (2019) focuses on the dynamic variant of the pick-up and delivery problem which uses capacitated dedicated vehicles and ad-hoc drivers like occasional drivers. The problem follows time windows where the dedicated vehicle route delivers the orders within the guaranteed time, some orders are crowdsourced by ad hoc drivers following the time constraints. This article models the dynamic settings where new orders and availability of ad hoc drivers occur throughout the day. The service platform takes this information and at each interval produces a match between the order and driver (either regular or OD). They propose a horizon rolling framework that repeatedly solves this matching problem of order, dedicated vehicle, and OD each time a new order or driver arrives. The performance of this model is compared with orders served by only dedicated vehicles and the results show that it is cost-effective to use OD along with dedicated vehicles and there is a huge influence of OD time flexibility and willingness on the performance of the system[9].

The study by Zhang et al. (2022) formulates truck delivery as a bi-partite hyper-matching problem. Here the shipper requires trucks to deliver goods and the carrier supplies the required trucks where the number of trucks supplied by the carrier either completely (an in-separable case) or partially (a separable case) fulfill the requirement of trucks. The incoming orders to the carrier are known over time and a history set of past orders is used to predict their probability distribution. In non-separable cases, the article provides a hyper-matching algorithm where the hyper-match is between the requirement of the order and the number of trucks served forming a hyperedge between the order vertex and truck vertex and the competitive ratio is a non-decreasing function of the history information on the number of orders. In separable cases, they provide an extended version of the algorithm named separable hyper-matching algorithm which can form a new hyperedge for the unmatched vertices of the order requirement and then add to the existing hyper-matching i.e., if the carrier is not able to fulfill the order from the shipper, it can partially be crowdsourced other trucks into the current hyper-matching problem. The proposed solutions are greedy heuristics to achieve fast results and batch-processing heuristics to get the best competitive ratio concluded by performing experiments on both synthetic and real-world datasets[10].

The article by Fatehi et al. (2022) aims for fast delivery service and assured delivery with short windows time performed by either the third-party logistics (3PL) who charge the cost of delivery per package or the crowd drivers who get paid an hourly wage for the distance traveled to do the delivery. Here, the incoming demand of customers is dynamic and follows the Poisson process which captures the random arrival of a customer's order. This influences the assignment of the delivery follows a random allocation policy and affects the compensation paid to ODs. The proposed approach is robust optimization and provides a closed-form solution to obtain insights on how the operations for such a system should be designed. The results show that using this approach there is a significant decrease in delivery costs with guaranteed service provided to the on-demand customers' orders[11].

Torres et al. (2022) study is an open variant of VRP that has stochastic destinations i.e., it consists of a single depot and fleet of dedicated vehicles that return to depot after delivery and capacitated occasional drivers (OD) with unknown destination locations. Here the customers are grouped into sectors, their orders are assigned to the vehicles (either dedicated or occasional), routes are performed to deliver the orders and the platform does these tasks in a two-stage stochastic modeling framework. Initially, routes of the vehicles are formed knowledge of the availability of ODs along with the compensation paid to them. Later the OD either accepts or rejects the order then a recourse action takes place by the dedicated vehicle in case of rejection of the order by OD. To improve the probability of OD accepting an order, historical data on OD's destination is used to create a discrete probability distribution to know the past number of orders accepted by OD per sector. The author proposed a branch and price algorithm and a column-generated heuristics for the quick generation of solutions. The computational results show that the OD capacity increases then the requirement of OD decreases and to maintain a high number of ODs, the capacity of each OD must be reduced. The route duration and capacity constraints of OD are directly proportional[12].

Silva et al. (2023) research work considers stochastic and dynamic environment settings which include a fleet of company vehicles and occasional drivers who deliver orders on the way to their home and the arrival of customer orders and the availability of ODs is random throughout the day within fixed time windows and uses deep reinforcement learning (DRL) approach which is data-driven i.e., there is a large set of historical scenarios available to train the neural network (NN). The decision is made in a two-stage approach, the first stage is solving an offline problem to assign routes to dedicated vehicles before the delivery is initiated and in the second stage at each scenario of the arrival of orders and available ODs are revealed, routes follow the first decision ordering, skips customers with no order and few orders are outsourced to the available ODs. The objective of the study is to minimize the total cost in the long term by assigning ODs to orders in optimal scenarios and paying small compensation to them. The computational results show that optimal routes are calculated as a recourse action for each scenario reveal[13].

The work by Tao et al. (2023) shows an online-to-offline (O2O) platform for the delivery of online orders using crowdsourced drivers that include their delivery capacity and delivery speed to propose a personalized dispatch model for order assignment and routing decisions. This model uses a feature set that includes the characteristics of the drivers and orders from past deliveries done and forecasts the speed of the driver on each trip and their delivery capacity that influences the number of orders on the trip. Two machine-learning models are used to personalize the forecast of the delivery speed and customize the delivery capacity and integrated with an optimization model to produce personalized order assignment and routing decisions. Here the feature set includes the behavior of the OD like their working experience, delivery performance, and geographical area knowledge. Also, the order and environment characteristics like the temperature, and the distance between the orders and the pick-up points are included in the feature set that trains the machine learning models to categorize the drivers based on their features. The computations done on the real-world dataset using the proposed model show that there is a significant reduction in the average delay of the orders[14].

The article by Silva et al. (2023) is a variant of stochastic capacitated VRP that incorporates the marginal distribution of the uncertainties i.e., the availability of ODs and the arrival of orders. The probability distribution of customers is known from historical data which also includes the constraints of the OD like capacity, availability, and routes they served in the past. The company uses ODs to help its fleet of vehicles in delivering the customers' orders to minimize the average total delivery cost concerning the worst-case probability distribution. The optimization is done in a two-stage approach: the first stage defines a solution to the offline problem i.e., to create routes ahead of time using historical data under the worst-case probability scenario. The second stage is the recourse action that takes place before the dispatch when the actual scenario is revealed. In the second stage after the reveal, the company will again information about the orders present, schedule the routes, and assign them to either the dedicated vehicle or the OD. Here the joint distribution is expressed as vectors of customers with no delivery order and customers with at least one available OD for all possible scenarios and each vector has a known marginal probability distribution. The study implements a branch-cut and price algorithm and uses column-dependent rows to iteratively add new routes and scenarios and provide better solutions. The computational results of the heuristics approximation show improvement in the time performance for large instances[15].

Simoni et al. (2023) work uses crowdsourcing of available couriers to deliver food orders from restaurants to customers in a dynamic setting i.e., the food orders are placed throughout the day and the courier can sign-in and sign-out any time of the day. So, the goal of the food delivery platform is to batch the orders and assignment to the courier in an efficient way such that a courier can serve orders in multiple pick-up and drop-off routes. The article proposes an order batching and assignment algorithm and a rolling horizon approach to decompose the problem into a series of sub-problems that are solved at small intervals of time. The approach has a basic and two advanced policies to improve the solutions over time. The myopic policy includes the current available information about the number of orders, and couriers and creates routes by using clustering, local search, and re-combine algorithms. The swap policy and insertion policy are used to re-optimize the solutions of myopic to increase the batch size and assignment of them to couriers to accommodate more orders in each route. At each re-optimization step, information about the arrival of new orders and couriers is considered to provide better and more sustainable routes for delivery. The analysis of results indicates that aggregating the order for courier shows a low decrease in the performance and yields cost savings due to the increase in batch size[16].

The article by Mancini et al. (2022) describes a technique to bundle incoming orders and assign them to an occasional driver and the model is named Vehicle Routing Problem- Occasional Driver- Order Bundle (VRP- OD- OB). This project consists of a single depot and multiple customers requesting orders, the orders can either be delivered by the company's fleet or by an occasional driver, who appears dynamically and the goal is to minimize the total cost which includes the travel cost of company's fleet route and compensation paid to the occasional drivers. The bundling of orders and their assignment is done in three stages: first where the company forms batches of orders depending on the locations of customers, then the available ODs will bid on the bundle of orders based on their willingness, and lastly, the company decides which bids to accept and routes are planned for company's vehicles to serve the orders that are not included in the bidding process. To build an attractive bundle of orders so that the bundles get bided by occasional drivers, the article uses two types of approaches namely cluster-based and corridor-based bundling and employees Large Neighbourhood Search, a metaheuristic to obtain the solution. The computational studies are done on two sets of randomly generated instances, the two approaches are tested and finally, the articles state corridor-based bundling shows better quality results since it uses the location of occasional drivers to create a bundle of orders rather than just using locations of customers. Thus, from this article, it is understood that the VRP problem is extended for the incorporation of Occasional Drivers and to group the orders into a bundle before assigning a company vehicle or an occasional driver[17].

The research work by Dayarian et al. (2020) uses the in-store customers as occasional drivers to deliver the received online orders in addition to company drivers to minimize the total cost which includes the company driver travel cost and compensation paid to the in-store customers for the online order delivery. Here, a single store is considered, both the in-store customers and online orders arrive randomly throughout the day and two rolling horizon approaches are specified where one uses the system's current information to make decisions while the other approach uses probabilistic information about the future to make the decisions. The in-store customers will notify their destination location and willingness after they arrive at the store, and the platform should decide to assign the online orders to the available in-store customer and/or the company drivers to reduce the total cost which is the combination of company vehicles' travel cost and compensation paid to in-store customers acting as occasional drivers. The compensation fee includes a fixed amount and the extra time OD needed to deliver the online order and company vehicle routes are formed by tabu search. The instances generated in this project are random for a geographical area of 100*100 square and computational study is done using different arrival rates of orders and customers on the static and dynamic variants of the model. In conclusion, the results state that in-store customers as ODs will be beneficial in terms of cost savings[18].

Tao et al. (2023) article proposes a pick-up and delivery problem with multiple depots where the system has full-time regular drivers, orders from customers are with time windows that arrive throughout the day, and occasional drivers who dynamically announce their availability over the time horizon along with their time windows. An event-based rolling horizon approach is proposed that iteratively runs a re-optimization procedure to update the delivery plan at each decision epoch which is triggered by the arrival of an occasional driver or when a prefixed time is reached. To fulfill the demand of customers, the system assigns the orders to either regular drivers or occasional drivers while considering the destination locations of ODs to minimize the total cost and total penalty cost for violation of customers' and ODs' time windows. The model builds a delivery re-plan at each decision epoch by re-optimizing the order assignments and route decisions based on newly available information about the available ODs and incoming orders. The instances used for this project are taken from Cordeau et al. (2001) research paper and computational studies are done to show the effectiveness of the proposed model and conclude that the use of ODs greatly reduces the stress on the company's delivery system[19].

2.3 Vehicle Routing Problem and Traveling Salesman Problem

To develop a foundational understanding of Mixed Integer Linear Programming (MILP) and time window constraints within the context of vehicle routing problems the following articles provided in detailed information.

The paper by Bae et al. (2016) is a variant of VRP with time windows. It is named a multi-depot vehicle routing problem with time windows and it is used to determine the optimal set of vehicle routes to satisfy the demand of customers within time windows from depots at different locations. A mixed integer programming model is developed to minimize the total cost incurred for the delivery of goods from depot to customer. A heuristic algorithm is proposed for the routing of the vehicle which is like nearest neighbor heuristics and the instances used are considered from Cordeau et al. (2001) article. This paper concentrates on modeling the delivery system with multiple depots and customers with a time window as a MILP problem. It also proposes a genetic algorithm to find good assignment and vehicle routes[20].

Calvo's (2000) article shows a new heuristic method for the Traveling Salesman Problem with Time Windows (TSPTW) where the problem consists of a single depot and multiple customers with time windows to minimize the total travel time. The algorithm consists of two phases, first is an assignment problem with relaxed scheduling constraints which provides a main tour and a small set of sub-tours starting and ending at the depot, followed by a greedy insertion procedure which is used to insert sub-tours to the main path and it is not possible another route is formed. The second phase is local search which is used to reduce the number of routes and improve the solution. The results of this article show that the new heuristic produces optimal or near-optical solutions for most of the cases within a reasonable amount of time[21].

In the food delivery industry, it is crucial to deliver the orders within a specific period, the research papers of Bae et al. (2016) and Calvo (2000) provided a clear understanding of vehicle routing problems with time windows, which helped to develop the delivery MILP problem. Some of the articles used a single depot to dispatch the orders, but articles like Dahle et al. (2019), Bae et al. (2016), Mancini et al. (2022) and Tao et al. (2023) used multiple pickup locations and their work helped to build this model, since it incorporates multiple restaurants that deliver the food orders. The involvement of external agents (namely occasional drivers) along with the company's fleet in the delivery of orders is introduced in Archetti et al. (2016) under a deterministic setting, i.e., complete information about the order and drivers is known in advance. Later this work was improved by Dahle et al. (2019) and Zhen et al. (2022), who have used the concept of crowdshipping to deliver the products by suggesting compensation schemes. But in real time, the online orders are dynamic which is used in our food delivery model. The rolling horizon approach is a popular method used to handle such orders and the papers Dayarian et al. (2020), Simoni et al. (2023), Tao et al. (2023), and Arslan et al. (2019) have used it and it is incorporated into this model and a decision epoch occurs whenever an order arrives or when a delivery resource is available. The concept of crowdshipping is included in the food delivery system and the incoming orders are grouped before assigning a delivery resource, which is taken as a reference from the work of Simoni et al. (2023), and Mancini et al. (2022). Penalty is imposed on the model for any late deliveries and a heuristic is developed to create routes for dedicated vehicle delivery. The concept that makes this model unique from previous work is the implementation of transfer delivery by an occasional driver to deliver a food order from a restaurant to a customer. This makes the model reduce the travel cost and utilize the occasional drivers for efficient delivery.

Table 2.1: Summary of the articles studied

Paper	Uncertainty	Orders from Customers	ODs
Archetti et al. (2016)	Deterministic	Known set of customers	Known set of ODs
Dahle et al. (2019)	Deterministic	Known set of customers	Known set of ODs
Zhen et al. (2022)	Deterministic	Known set of customers	Known set of ODs
Parappathodi and Archetti (2022)	Deterministic	Known set of customers	Known set of ODs
Alp M. Arslan et al. (2019)	Dynamic	New order arrival known over time	OD is available known over time
Zhang et al. (2022)	Dynamic	New order arrival known over time	OD is available known over time
Fatehi et al. (2022)	Dynamic	Customers arrival changes	OD route duration and compensation paid changes accordingly
Torres et al. (2022)	Stochastic	Known set of customers	OD destination is random
Silva et al. (2023)	Stochastic	Random	Random
Tao et al. (2023)	Dynamic	New order arrival known over time	Delivery capacity and delivery speed change according to the driver
Silva et al. (2023)	Stochastic	Random	Random
Simoni et al. (2023)	Dynamic	New order arrival known over time	OD is available known over time
Mancini et al. (2022)	Dynamic	Known set of customers	OD is available known over time
Dayarian et al. (2020)	Random	Random	Random
Tao et al. (2023)	Dynamic	New order arrival known over time	OD is available known over time
Bae et al. (2016)	Deterministic	Known set of customers	Known set of ODs
Calvo (2000)	Deterministic	Known set of customers	Known set of ODs

3
Problem Description

The region is described as a graph $G = (N, A)$ where N is the node set and A is the arc set. The node set represents the locations of the restaurants, customers, and occasional drivers, where at each restaurant a dedicated vehicle is present. The arc set represents links connecting the nodes. The node set N is defined as follows:

$$N = R \cup C \cup V \cup D$$

where R is the set of restaurants, C is the set of customers, V is the set of dedicated vehicles (one at each restaurant), and D is the set of occasional drivers.
The arc set A is defined as follows:

$$A = A_1 \cup A_2 \cup A_3$$

where each subset of arcs is defined as follows:

- $A_1 = \{(v, j) \mid v \in V, j \in C\}$ represents dedicated vehicle deliveries from restaurants to customers by dedicated vehicles.
- $A_2 = \{(d, j) \mid d \in D, j \in C\}$ represents direct deliveries from restaurants to customers by occasional drivers.
- $A_3 = \{(d, d', j) \mid d, d' \in D, d \neq d', j \in C\}$ represents transfer deliveries by two occasional drivers to deliver the orders.

We consider an online food delivery platform whose goal is to deliver food orders from restaurant to customer within their time windows in the most cost-effective manner. Here the platform receives continuous food orders over time, where each order has time windows. Let e_j be the *earliest departure time*, which is the earliest time that an order j can leave the restaurant, and l_j be the *latest arrival time*, which is the latest time that an order can be received by the customer j. Let t_j be the time taken to travel from restaurant to customer j, where $j \in C$. Therefore, the order must be received by the customer by *latest arrival time* i.e., $t_j \leq l_j$, otherwise a penalty is imposed on the platform. Let a_j be the announcement time of order j and $\bar{l}_j$ be the *latest departure time* and $\bar{l}_j = l_j - t_j$. As mentioned in the definition of arc set, an order can be delivered by a restaurant's dedicated vehicle (DV), or by occasional driver (OD) delivery, or by transfer delivery using two occasional drivers. Here, each occasional driver has a detour radius, r_d which is the area that an occasional driver can serve. It can be noted that often in this paper order, j refers to a customer, where $j \in C$. The figure 3.4 displays the announcement time and the associated time windows of an order, j.

The delivery of a food order from a restaurant to the customer by a dedicated vehicle incurs a travel cost, c_{vj} which is equal to the Euclidean distance between the restaurant and the customer as shown in figure 3.1. The occasional driver can directly deliver an order, when the location of the restaurant and the location of the customer are within its detour radius, and a fixed compensation fee, f is paid for the delivery which is smaller than the travel cost as shown in figure 3.2. In a transfer delivery scenario, the location of the restaurant must be within the detour radius of one occasional driver, the location of the customer must be within the detour radius of another occasional driver and the detour radii of both the occasional drivers must intersect, and a fixed compensation fee, p is paid to each occasional driver for the transfer delivery which is also smaller than travel cost as shown in figure 3.3. Here, the transfer of parcel from one occasional driver to another happens on the plane of intersection of detour radii of both the occasional drivers. Each occasional driver can perform up to one direct delivery and one transfer delivery per trip. The dedicated vehicle can carry as many orders as possible and its capacity is not limited. The parameters used in this model are written in the table 3.1. The journey of a dedicated vehicle starts at its associated restaurant, serves the assigned customers, returns to its restaurant and DV is ready for the next trip. Similarly, the journey of an OD starts at the restaurant, delivers and/or transfers the order to the customer and/or another occasional driver, depending on the assignment of orders and the OD is ready for the next trip.

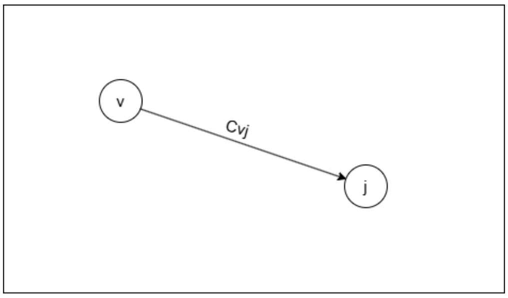

Figure 3.1: Delivery by DV, v to Customer, j

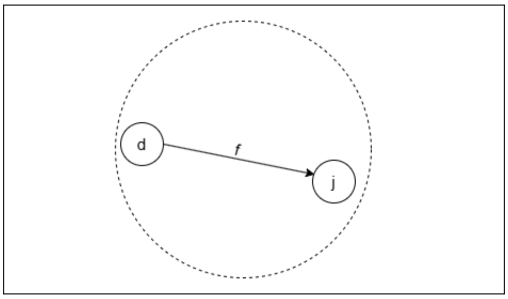

Figure 3.2: Delivery by OD, d to Customer, j

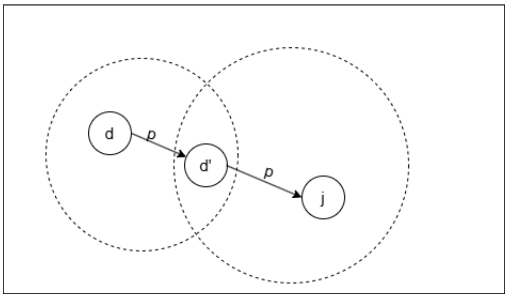

Figure 3.3: Transfer Delivery by ODs, d and d' to Customer, j

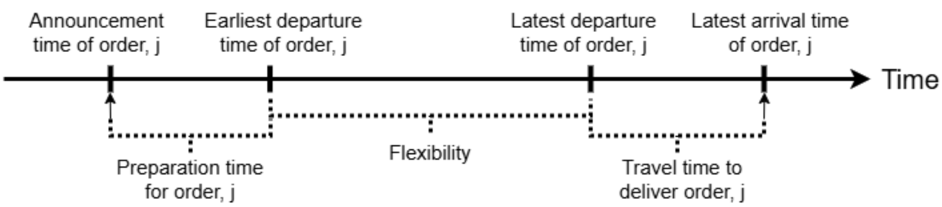

Figure 3.4: Time windows associated with an order, j

Table 3.1: List of Parameters

Parameter	Description
a_j	Announcement time of order j
e_j	Earliest departure time of order j
l_j	Latest arrival time of order j
t_j	Travel time to order j
$\underline{l}_j$	Latest departure time of order j
c_{vj}	Travel cost by dedicated vehicle v to customer j
r_d	Detour radius of occasional driver, d
f	Compensation fee in direct delivery by OD
p	Compensation fee in transfer delivery by OD

Table 3.2: List of Decision Variable

Decision Variables	Description
x_{vj}	Binary variable is 1 if $(v, j) \in A1$ is active
y_{dj}	Binary variable is 1 if $(d, j) \in A2$ is active
$z_{dd'j}$	Binary variable is 1 if $(d, d', j) \in A3$ is active
b_{vj}	Continuous variable for arrival of DV, v at customer, j
b_{dj}	Continuous variable for arrival of OD, d at customer, j

The decision variables used in the model are mentioned in the table 3.2. In this model, there are three binary decision variables and two continuous decision variables. The binary variable, x_{vj} is 1 when dedicated vehicle, v delivers the order to customer, j, otherwise it is zero. The binary variable, y_{dj} is 1 when occasional driver, d delivers the order to customer, j, otherwise it is zero. The binary variable, $z_{dd'j}$ is 1 when a transfer is done between occasional drivers, d and d' to deliver the order to customer, j. The continuous variable, b_{vj} is the value representing the arrival of dedicated vehicle, v to customer, j location, and the continuous variable, b_{dj} is the value representing the arrival of occasional driver, d to customer, j location.

Objective Function

The objective is to minimize the total cost. The total cost is the sum of three components, one is the travel cost incurred by the dedicated vehicle deliveries, second is the total compensation paid to occasional drivers for direct deliveries and lastly, the total compensation paid to occasional drivers for transfer deliveries. The equation 3.1 describes the components of total cost and aims to minimize it.

$$\text{Minimize} \quad \sum_{(v,j)\in A_1} c_{vj}x_{vj} + \sum_{(d,j)\in A_2} fy_{dj} + \sum_{(d,d',j)\in A_3} 2pz_{dd'j} \tag{3.1}$$

Constraints

- Exclusive assignment of each order:

$$\sum_{v\in V} x_{vj} + \sum_{d\in D} y_{dj} + \sum_{\substack{d,d'\in D\\ d\neq d'}} z_{dd'j} \leq 1 \quad \forall j \in C \tag{3.2}$$

  The above equation 3.2 states that each order must be assigned to only one mode of delivery i.e., either by dedicated vehicle or by occasional driver, or by transfer between occasional drivers.

- Capacity of dedicated vehicle: A dedicated vehicle has unlimited capacity i.e., each dedicated vehicle can possibly deliver all the incoming orders received by the restaurant.

- Capacity of the occasional driver: An occasional driver can handle at most one direct delivery at a time i.e., once an occasional driver, d is given order as direct delivery, it can not be assigned to deliver any other order in direct delivery till d finishes the direct delivery as shown in equation 3.3.

$$\sum_{j\in C} y_{dj} \leq 1 \quad \forall d \in D \tag{3.3}$$

  A pair of occasional drivers can be involved in atmost one transfer delivery at a time i.e. when d and d' occasional drivers are paired as transfer delivery, d can not be assigned to deliver any other order in transfer delivery till it finishes the transfer of the order to d' and d' can not be assigned to deliver any other order in transfer delivery till it finishes the delivery of received order to the customer as shown in equation 3.4.

$$\sum_{j\in C} (z_{dd'j} + z_{d'dj}) \leq 1 \quad \forall d, d' \in D, d \neq d' \tag{3.4}$$

- The time window constraints are given as follows: If x_{vj} = 1, then

$$l_j \geq b_{vj} \geq t_{ij} + e_j \quad \forall v \in V, j \in C \tag{3.5}$$

The above equation 3.5 states that when x_{vj} is active, the arrival of v dedicated vehicle at j customer must be within the time window of the order.

If y_{dj} = 1, then

$$l_j \geq b_{dj} \geq t_{ij} + e_j \quad \forall d \in D, j \in C \tag{3.6}$$

The above equation 3.6 states that when y_{dj} is active, the arrival of d occasional driver at j customer must be within the time windows of the order.

If $\sum\limits_{\substack{d \in D \\ d \neq d'}} z_{dd'j} = 1$, then

$$l_j \geq b_{d'j} \geq t_{ij} + e_j \quad \forall d', j \in C \tag{3.7}$$

The above equation 3.7 states that when $z_{dd'j}$ is active, the arrival of d' occasional driver at j customer must be within the time window of the order.

This delivery problem aims to determine the assignment of food orders to an occasional driver or a dedicated vehicle or to a pair of occasional drivers such that the delivery platform can minimize the total cost. In this model, it is assumed that the dedicated vehicle can carry as many orders as possible i.e., its capacity is unlimited. All the incoming orders need to be served by either a dedicated vehicle or an occasional driver and no order is rejected. Once a dedicated vehicle is assigned a route, another route can be assigned to that dedicated vehicle when it returns to the restaurant.

4
Solution Approach

The customers place their food orders dynamically at the delivery platform and these orders arrive throughout the day when an order arrives, it is crucial to check the availability of occasional drivers that can deliver the food orders. Therefore, an event-based rolling horizon approach is proposed to repeatedly solve the above-mentioned MILP (mixed integer linear programming) delivery problem and determine the best assignment of order with the available information which includes the dedicated vehicle and occasional drivers that have just finished their previous tasks and are ready accept new orders. and plan the delivery route for the dedicated vehicle if any orders are assigned to it.

A rolling horizon is a planning and decision-making approach that can be implemented in a dynamic food delivery system to manage the assignment of orders that arrive over time. It is used to make decisions for a fixed future period and update the plan by incorporating new information and adjusting the decisions accordingly as time progresses. This approach continuously adapts to changing conditions and balances the current actions and the main objective of the system. Along the time horizon, decisions are made at certain intervals of time known as decision epochs. At each decision epoch, the current information in the system is assessed and a decision is made that will influence future actions. In our model, the decision epoch occurs when a certain number of orders have arrived and when a delivery resource is available and this criterion is explained in detail in the later part of the section. The figure 4.1 shows the overview of the model displaying the steps involved in the delivery platform.

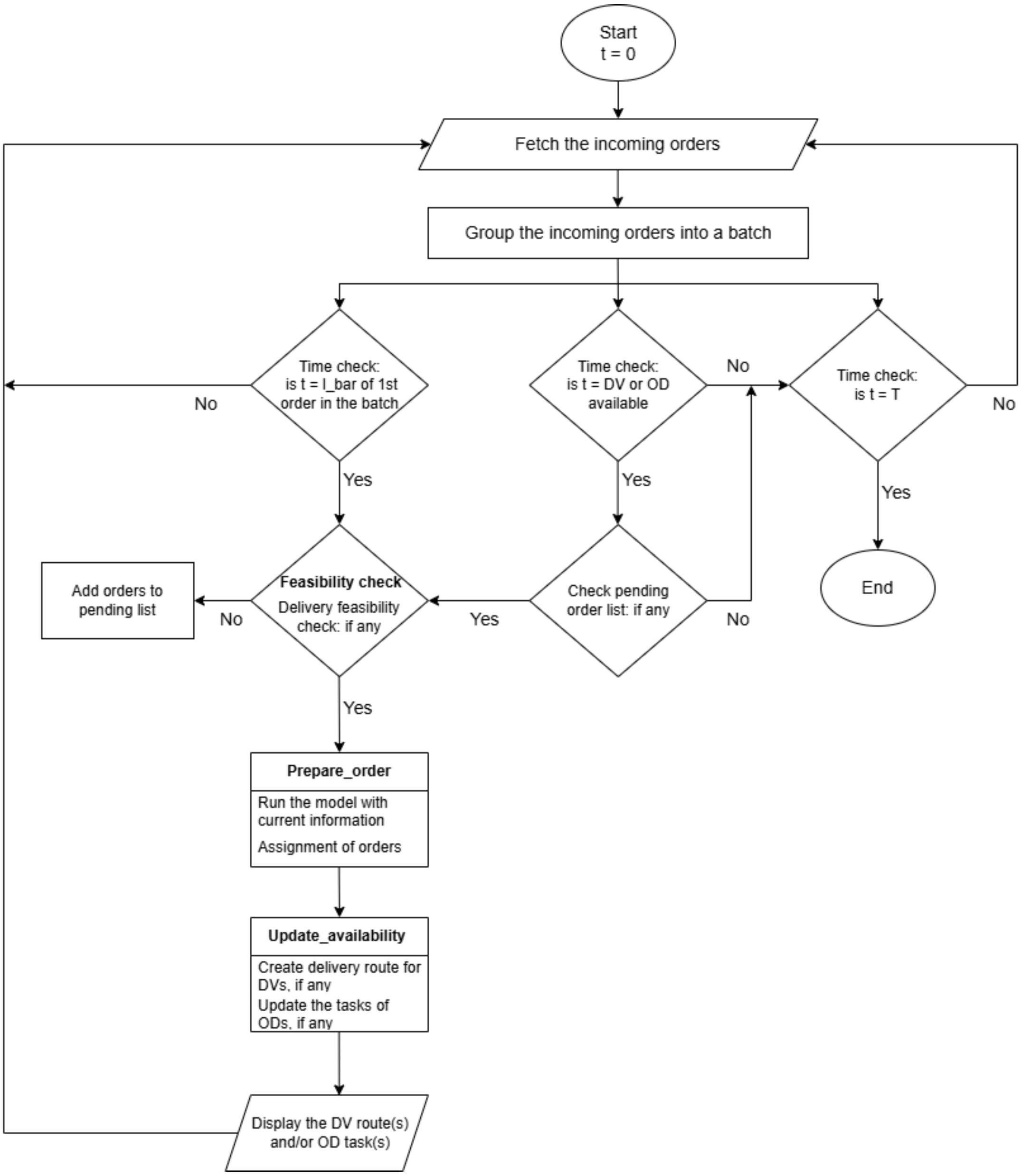

Figure 4.1: Food Delivery System Framework

The process begins with the initialization of time at t = 0, marking the start of the delivery system's operation and the orders arrive dynamically over the time horizon, T. The first step involves fetching the incoming orders and each order has a corresponding time window. As orders arrive, they are grouped into a batch for efficient processing. The platform waits until it has enough orders to make batch processing, which allows for optimization in the delivery routes and resource allocation. As the orders are grouped, the system performs a time check to determine if the current time, t has reached the latest departure time, $\bar{l}$ of the first order in the batch. If the time is not yet $\bar{l}$, the system loops back to fetch and include more orders into the batch. If the time has reached $\bar{l}$, the decision epoch is triggered and the batch of orders is sent to the 'feasibility check' module. This module receives a set of orders, and it is used to determine the feasible occasional drivers and dedicated vehicles that are available to deliver each order in the batch. An order can be delivered in three ways, dedicated vehicle delivery, occasional driver delivery, and transfer occasional driver delivery. If the dedicated vehicle present at the restaurant is available, then for that order dedicated vehicle delivery is feasible. An occasional driver delivery is feasible if the locations of the restaurant and customer are within the detour radius of that occasional driver. If the location of the restaurant is within the detour radius of one occasional driver, the location of the customer is within the detour radius of another occasional driver and if the detour radii of the two occasional drivers intersect, then this pair of ODs is feasible to deliver the order in the transfer delivery mode. Therefore, this module is responsible for segregating the orders and determining which delivery resource is feasible for each order in the batch. If an order does not have a feasible delivery resource (i.e., DV or OD or transfer between ODs), then that order is sent to the list of pending orders. The next step is to send this batch of orders and corresponding feasible delivery sets to the optimization model.

The module named 'prepare_order' runs the optimization model described in the previous section. In this module, all the required input parameters, batch of orders, and feasible delivery sets are given as input. The decision variables, the objective function, and the necessary constraints along with the time windows of the orders mentioned in the previous section are specified in this module. The model runs with this information and the objective to minimize the total cost. The output of this module is the assignment of the order to either occasional driver delivery occasional driver transfer delivery or dedicated vehicle delivery. At the end of this module, if any of the dedicated vehicles are used to deliver an order, then it is important to decide the delivery route of that dedicated vehicle. Thus, the output assignment of orders is given to a routing algorithm which plans the routes for the dedicated vehicles and handles late deliveries if any.

The module named 'update_availability' runs the routing algorithm for the dedicated vehicle and organizes the tasks of occasional drivers depending on the order assignment result. A single occasional driver can be assigned to direct delivery transfer delivery or both. The dedicated vehicle and the corresponding set of customers that it needs to deliver are given as input to the routing algorithm. The route for a dedicated vehicle starts at the restaurant serves the customers and returns to the restaurant. It is important to serve the customer according to their time windows, and the sequence of visiting the customers is scheduled by the routing algorithm. It is considered that the dedicated vehicle leaves its restaurant to serve the customers only after the preparation of all the orders assigned to the dedicated vehicle. The routing algorithm initially sorts the list of customers according to their earliest departure time and then inserts each customer into the route using the nearest neighbor search. As the output, the routing algorithm calculates the total cost and total time taken for each route. If there are any late deliveries, then a penalty is added to the total cost per late delivery to the customer. The steps involved in the routing algorithm are shown in figure 4.2. This module also updates the availability of the occasional drivers for their next deliveries by tracking the tasks assigned to the ODs. Here, an occasional driver can deliver an order in direct delivery and/or be involved in a transfer delivery, so the module displays the OD performs which tasks and adds a penalty for any late deliveries to the total cost. Then the display module represents the order assignment and vehicle routing in the form of a graph.

Routing Algorithm: *(Nearest Neighbour Search)*

```
Input: dedicated vehicle and list of customers
Output: delivery route for dedicated vehicle, route = [restaurant, list of customers in a
sequential path, restaurant]
dedicated vehicle's route starts from restaurant (= previous node)
while There are still customers in the list of customers do
        Insert the nearest customer (= current node) into the route of dedicated vehicle
        Calculate the travel time and travel cost from previous node to current node
        if travel time exceed latest departure time of current node then
            Add penalty to travel cost
        end if
        Remove that customer from the list of customers and previous node = current node
end while
dedicated vehicle's route ends at restaurant
```

Figure 4.2: Steps involved in Routing Algorithm

When a dedicated vehicle or occasional driver finishes their assigned task, they become available to deliver the next orders. Therefore, the delivery system tracks the timing when resources (either DV or OD) become available for the next delivery assignment. The return of a dedicated vehicle to the restaurant or when an occasional driver finishes their task(s), a decision epoch is triggered. At this decision epoch, the pending order list is checked, if the list is empty, then the resource waits and the system continues the fetch new orders. If the pending list has orders waiting for an assignment, then the feasibility check module is processed to know whether the available resource can deliver the pending order, if it is possible then the prepare_order module is called and the above-mentioned process is performed to deliver the order. If the available resource is not feasible to deliver the pending order, then the pending order waits for the next resource availability. The process includes a final time check to determine if the end time T has been reached. If the current time is not yet T, the system continues to operate. Once T is reached, the system ends, marking the completion of the delivery platform's operations for that period.

5

Computational Study

To study the performance of the proposed model, a series of computational experiments are carried out on a PC with AMD RYZEN-5000 processor, CPU 2.10 GHz, 8GB of RAM, 64-bit operating system, and using IBM ILOG CPLEX Optimization Studio version 22.1.1.0. The MILP delivery problem described in section 3 and the routing algorithm described in section 4 are implemented using Python Programming Language and the results are presented using CPLEX solver's default parameter settings. The following is the discussion of the computational performance and results of the proposed delivery model.

CPLEX is an optimization software developed by the IBM company and it is used to efficiently handle linear programming (LP), mixed integer linear programming (MILP), and quadratic programming (QP) problems. The solver's core strength is mathematical programming and it efficiently navigates the vast solution spaces of these problems using techniques such as branch-and-cut, heuristic methods, and advanced preprocessing to prune infeasible paths and reduce computational overhead. This makes it exceptionally good at finding optimal or near-optimal solutions within practical timeframes, and it is an excellent choice to handle the proposed MILP delivery model where decision-making speed is crucial.

5.1 Instances

A large set of instances is generated which comprises several instances, and each instance is characterized by the input parameters such as number of restaurants, number of dedicated vehicles per restaurant, number of occasional drivers, and number of customers given to the model. The time horizon, T of the model is set to 3 hours. In each instance, the customers, the restaurants, and the occasional drivers are randomly located in a geographical region represented by a 200 x 100 rectangle. These locations are randomly generated by NumPy library's random module which generates locations using uniform random distribution. This method of creating instances is inspired by the research papers by Dayarian et al.(2020)[18], Zhen et al.(2021)[7], Mancini et al. (2022)[22], and Silva et al.(2023)[13] who all have generated random test instances. It is assumed that the travel time (in minutes) and the cost (in monetary value) are deterministic and proportional to the Euclidian distances between restaurant and customers and this is adapted from the article by Silva et al.(2023)[13]. The compensation paid to occasional drivers is fixed which is also taken from the paper by Silva et al.(2023)[13]. Each customer order is associated with specific time windows that show the earliest and latest times by which the order must be delivered to the customer and the way of selecting the time windows and order arrival time is adapted from the research paper Alp M. Arslan et al.(2019)[9] which also specifies the flexibility of delivering an order by considering the latest departure of the order.

Here, the number of customers used are {25, 50, 75, 100}, the number of restaurants used are {5, 10}, the number of occasional drivers used are {20, 40, 80}, and at each restaurant the number of dedicated vehicles used are 1, 2, and 3. The output of different instances is compared using the following key performance metrics:

- *Order fulfillment rate:* It is defined as the percentage of the total number of orders delivered to the total number of orders placed.
- *On-time delivery rate:* It is defined as the percentage of total number of orders delivered within their time windows to the total number of orders placed.
- *Orders sent to Pending:* It is defined as the percentage of total number of orders sent to the pending list since no occasional driver or dedicated vehicle is available to make the delivery.
- *Total cost:* It is the entire cost to deliver all the orders

5.2 Offline Problem

The delivery model described in section 4 works in a dynamic environment, where information about the orders is revealed over time and the model uses a time horizon approach to get the solution. In this section, an offline problem is proposed to serve as a lower bound for the performance evaluation of the dynamic delivery model. The offline problem is a scenario, where the complete information about the orders and the occasional drivers is known in advance as a priori, and it produces an optimal solution. This offline problem is used to provide the best-case scenario since this model has all the information available upfront and decisions can be planned accordingly to obtain the best output. In this delivery system, the offline problem can be used to determine the least total cost possible if all customer orders throughout the day are known from the start. The offline model calculates the best possible order assignment and dedicated vehicle delivery routes.

The working of the offline problem is shown in figure 5.1. The offline problem has the same objective as the dynamic model which is to minimize the total cost to deliver all the orders. Since the model is a static type, all orders are grouped into a single batch and are sent to the *feasibility_check* module. This module will provide feasible occasional drivers for direct delivery and transfer delivery respecting their detour radius, and feasible dedicated vehicles considering the order's restaurant. The set of feasible deliveries is then sent to *prepare_order* module which runs the optimization code to assign the orders to either a dedicated vehicle or occasional driver or to a pair of occasional drivers to minimize the total cost. The offline problem assigns the delivery routes for the dedicated vehicles and handles the tasks of the occasional driver using the *update_availability* module.

The output of the offline problem acts as a benchmark for the dynamic delivery model. A comparison between the results of the offline problem and the dynamic model is shown in table 5.1. Different configurations are given as an input where R represents the number of restaurants, C represents the number of customers, D represents the occasional drivers, and each restaurant has a single dedicated vehicle. In terms of the objective value, the offline problem shows a lower objective value compared to the dynamic model across all configurations. Since the complete information is known, the offline model can be optimized more efficiently. The dynamic model operates with information revealed over time and its objective value comes close to the offline model's performance. A routing algorithm is used after the order assignment and

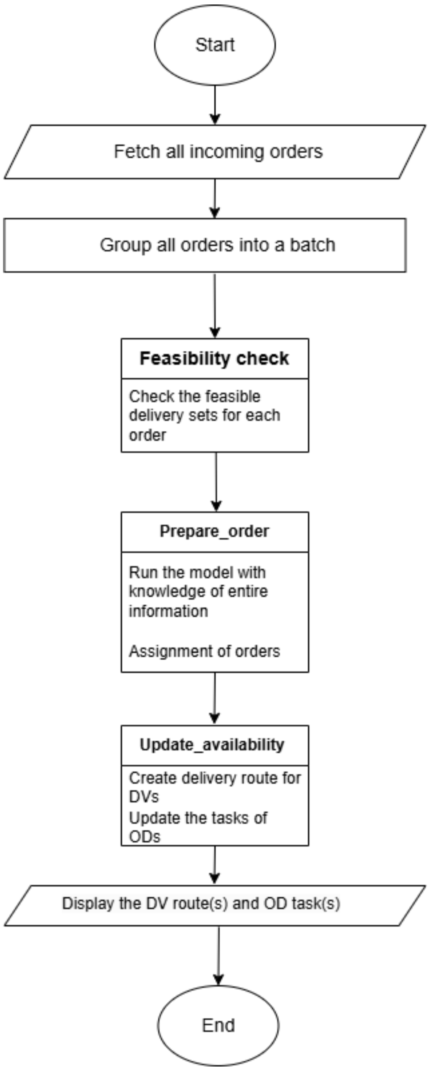

Figure 5.1: Working of Offline Problem

the total cost for the dynamic model is higher across all configurations which acts as an upper bound for each configuration. The offline model generally achieves a higher on-time delivery rate compared to the dynamic model. This is expected, as the offline model operates with full knowledge of all variables and constraints ahead of time. The on-time delivery rate for both models tends to decrease as the complexity of the configuration increases i.e., in case of high demand from customers and less occasional drivers and dedicated vehicles to deliver the orders. The offline model offers theoretical benchmarks for performance but the dynamic model reflects the challenges and realities of operational environments, where decisions must be made with incomplete information.

Configuration (R, C, D)	Objective Value		Total Cost		On-time Delivery Rate (%)	
	Offline	Dynamic	Offline	Dynamic	Offline	Dynamic
(5, 30, 20)	2978.027	3263.672	4040.671	4178.925	100	96.0
(5, 50, 30)	5297.448	5846.492	6913.572	7009.756	94.0	92.0
(10, 50, 20)	5268.308	6552.790	8279.526	8413.627	94.0	80.0
(10, 100, 40)	8457.105	10129.068	12285.851	12645.663	95.0	89.0
(15, 150, 60)	15223.630	16341.816	20961.735	21409.946	92.0	86.0

Table 5.1: Comparison of Offline and Dynamic Results

5.3 Computational Performance

This section analyses the computational performance of the proposed food delivery model. A sample image of the geographical region which includes the locations of restaurants, customers, occasional drivers, and their detour radius is shown in figure 5.2. It depicts the region with 10 customers, 5 restaurants each having a single dedicated vehicle as standby, and 5 occasional drivers. The figure 5.3 shows the delivery of food orders from restaurant to customers after running the delivery model and figure 5.4 displays the solution of the delivery model.

The table 5.2 and table 5.3 shows the output of different instances of the delivery model and the results in each scenario is obtained within a few minutes by using the CPLEX solver. In table 5.2, the number of restaurants is fixed to 5 and each restaurant has a single dedicated vehicle, then various combinations of several customers and occasional drivers are given to the model to analyze its performance. In this case, the order fulfillment rate remains at 100% for most cases except for scenarios where there are too many customers. For instance, 5 dedicated vehicles and 20 occasional drivers serving 100 customers lead to 56% order fulfillment. This shows that when the driver-to-customer ratio is low, order fulfillment suffers and the on-time arrival rate tends to decrease as customer numbers increase. For instance, with 5 dedicated vehicles and 20 occasional drivers serving 50 customers, the on-time arrival is 62%, and it further drops to 58.66% when serving 75 customers, which highlights delivery delays with a higher number

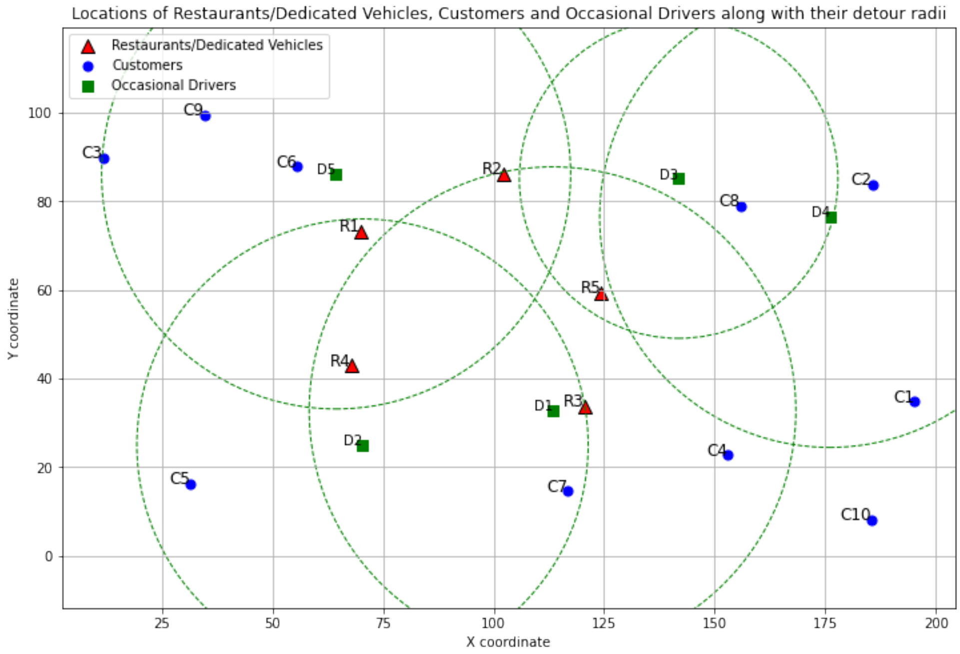

Figure 5.2: Region with Restaurants, Customers and ODs

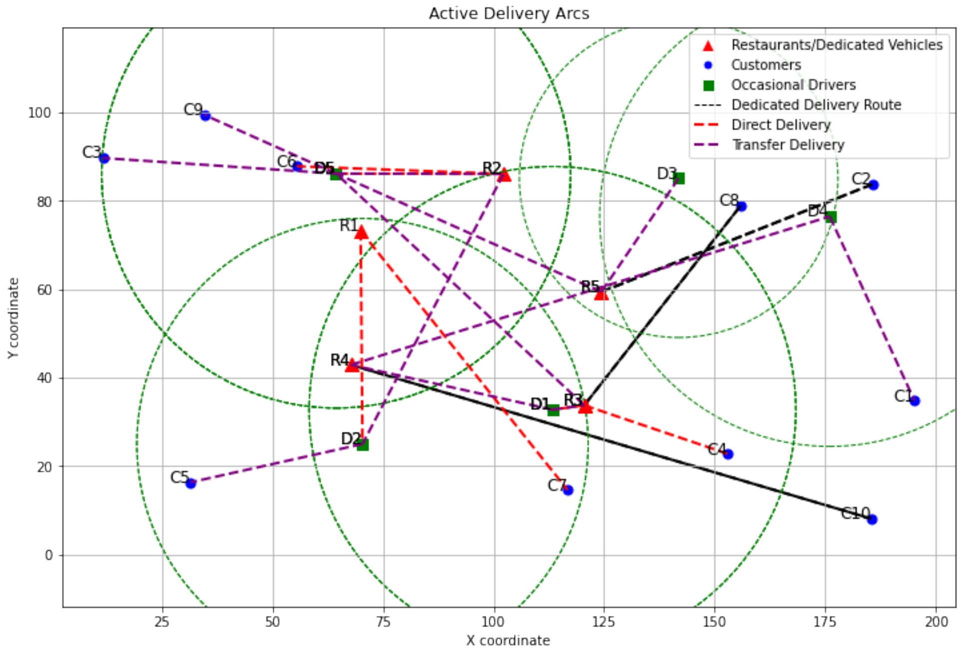

Figure 5.3: Delivery of Food Orders

```
**********The route by each dedicated vehicle is: {4: [4, 10, 4], 3: [3, 8, 3], 5: [5, 2, 5]}
Total time traveled per route: {4: 250.2102130524203, 3: 119.76327119926145, 5: 136.73987604649844}
Total cost per route: {4: 490.4204261048406, 3: 229.5265423985229, 5: 263.4797520929969}
Compensation paid to occasional drivers in direct delivery and transfer delivery: defaultdict(<class 'int'>, {2: 50, 5: 120, 1:
190, 3: 70})
 defaultdict(<class 'int'>, {2: 70, 5: 140, 4: 70})
Driver 2 route: [('transfer_receive', 5, 47, 52), ('direct_deliver', 7, 186, 191)]
Driver 5 route: [('direct_deliver', 6, 35, 40), ('transfer_receive', 9, 89.10045263764783, 94.10045263764783), ('transfer_sen
d', 5, 143.7811470598714, 143.7811470598714), ('transfer_receive', 3, 204.9471243108555, 209.9471243108555)]
Driver 1 route: [('transfer_send', 1, 37, 37), ('transfer_send', 3, 98.16597725098411, 98.16597725098411), ('direct_deliver',
4, 169, 174)]
Driver 3 route: [('transfer_send', 9, 42, 42)]
Driver 4 route: [('transfer_receive', 1, 37, 42)]
```

Figure 5.4: Output of the Delivery Model

Restaurants	Occasional Drivers	Customers	Total Cost (in euros)	Order Fulfillment (%)	On-time Arrival (%)	Orders sent to Pending (%)
5	20	25	3023.01	100	100	0.0
5	20	50	7593.71	100	62.0	50.0
5	20	75	10789.99	100	58.66	42.66
5	20	100	6466.63	56.0	80.35	44.0
5	40	25	3011.31	100	100	0.0
5	40	50	5970.06	100	96.0	0.0
5	40	75	9211.62	100	85.33	1.3
5	40	100	13984.43	100	77.0	4.0
5	80	25	2129.13	100	100	0.0
5	80	50	4483.90	100	96.0	0.0
5	80	75	7000.96	100	89.33	0.0
5	80	100	9314.32	100	94.0	0.0
5	80	200	22639.85	100	78.0	0.0

Table 5.2: Delivery Performance with 5 Restaurants

of customers and the same number of drivers. The total cost increases with more customers and the percentage of pending orders increases as the delivery resources are stretched with the increase in customers. When there are sufficient delivery resources relative to customers, like in the case of 80 occasional drivers for customer loads, there are no pending orders, and on-time arrivals are higher compared to other cases, reflecting optimal performance under these conditions. Therefore, the delivery system functions best with a higher ratio of drivers to customers, ensuring both timely deliveries and higher order fulfillment rates, while high customer loads with limited occasional driver availability lead to decreased order fulfillment, delayed arrivals, and increased pending orders.

The table 5.3 portrays a similar scenario with more restaurants in the region, i.e., increased from 5 to 10 restaurants, each having a single dedicated vehicle. This allows for a better distribution of delivery resources, especially under higher customer loads. It is evident in scenarios where

Restaurants	Occasional Drivers	Customers	Total Cost (in euros)	Order Fulfilment (%)	On-time Arrival (%)	Orders sent to Pending (%)
10	20	25	3028.10	100	100	0.0
10	20	50	7806.15	100	88.0	0.0
10	20	75	10765.46	94.66	81.69	24.80
10	20	100	6848.53	54.0	83.33	46.0
10	40	25	2683.06	100	100	0.0
10	40	50	5979.45	100	98.0	0.0
10	40	75	7784.38	100	94.66	0.0
10	40	100	12623.22	100	83.0	0.0
10	80	25	1670.92	100	0.0	0.0
10	80	50	4477.27	100	100	0.0
10	80	75	7070.26	100	98.66	0.0
10	80	100	8419.99	100	98.0	0.0
10	80	250	28151.70	100	80.0	0.4

Table 5.3: Delivery Performance with 10 Restaurants)

the number of occasional drivers remains the same, but the increase in restaurants leads to more balanced workloads. The on-time arrival rates in this scenario are improved when compared to the table 5.2. The total cost tends to be slightly increase, due to the dedicated vehicle deliveries from each restaurant, but this trade-off cost appears to be well-justified in terms of better fulfillment and on-time rates. This setup distributes delivery responsibilities more evenly, leading to better performance, particularly in high-demand scenarios. The two scenarios with 5 restaurants, 80 occasional drivers, 200 customers, and 10 restaurants, 80 occasional drivers, and 250 customers show the maximum customer load each scenario can handle and provide a feasible solution with good quality and within a reasonable amount of time (approximately 3 minutes).

Restaurants	Dedicated Vehicle(s)	Occasional Drivers	Customers	Total Cost (in euros)	Order Fulfilment (%)	On-time Arrival (%)	Orders sent to Pending (%)
5	1	20	100	6466.63	56.0	80.35	44.0
5	2	20	100	8933.81	73.0	42.46	27.0
5	3	20	100	12400.63	100	47.0	0.0
10	1	20	100	6848.53	54.0	83.33	46.0
10	2	20	100	9850.61	78.0	58.97	38.0
10	3	20	100	12674.35	100	56.99	0.0

Table 5.4: Delivery with Dedicated Vehicles

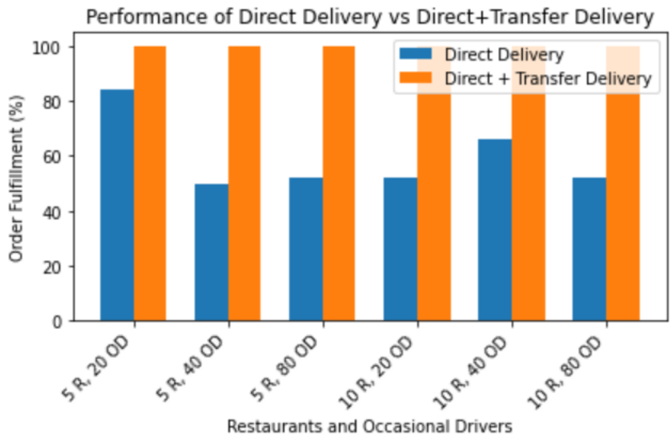

Figure 5.5: Comparison between OD direct delivery and transfer delivery

The table 5.4 draws a few crucial insights, where instead of using a single dedicated vehicle at the restaurant, more than one dedicated vehicle are used which shows a scenario to use more dedicated vehicles to meet the customer demand. From the results of this scenario, it is evident that adding more dedicated vehicles incurs high costs without substantial improvements in on-time delivery. Occasional drivers are paid smaller compensation. Therefore, a more efficient and cost-effective approach would be to increase the number of occasional drivers, who can handle the delivery load more flexibly and at a lower cost and improve the overall performance of the delivery model, particularly in terms of on-time deliveries.

The figure 5.5 displays the performance of the delivery model by using different delivery methods for occasional drivers i.e., direct delivery by a single one and transfer delivery by a pair of occasional drivers. The results of solely using direct delivery of occasional drivers in the delivery model are given in Appendix A. The results of the graph show that using a combination of direct and transfer deliveries is far more efficient than relying solely on direct delivery. This approach allows for 100% order fulfillment across various scenarios, making it a highly effective strategy for ensuring all orders are completed, even when occasional drivers face increased demand. By utilizing transfer delivery, the system optimizes the use of occasional drivers leading to better performance and customer satisfaction without requiring an excessive number of drivers.

6
Conclusion and Future Research

This paper explored a form of crowdshipping that uses transfer between occasional drivers which supplements the direct delivery by occasional driver and delivery by dedicated vehicle to fulfil the online food orders, a novel concept in food delivery system to provide quick and low-cost home delivery for food orders placed online. The delivery platform considers the standby dedicated vehicles at the restaurants and a set of occasional drivers volunteering to perform the food order delivery in return for a fixed compensation. A MILP problem is formulated to assign the food order from the customer to either a dedicated vehicle or an occasional driver or a pair of occasional drivers to reduce travel costs and a routing algorithm is built using nearest neighbor search to create routes for the dedicated vehicles that received the orders. An event-based rolling horizon approach is proposed to handle the real-time arrival of the orders and updates of the occasional drivers, which re-optimizes the delivery system whenever new information is available. This approach solution is compared to the offline problem, which acts as a benchmark for the dynamic delivery model and the results show that the output of the dynamic model is close to the offline problem.

The computational tests are performed on the model using different instances that depict real-time scenarios with varied numbers of restaurants, customers, dedicated vehicles, and occasional drivers. The results show that the proposed model adapts to the changing conditions, and the use of occasional drivers in both ways (direct delivery and transfer delivery) helps to provide better results in terms of order fulfillment, maintaining lower travel costs, and reducing delivery delays. Furthermore, the study shows the scalability of the delivery model showing that with an increase in orders or occasional drivers, the system provides efficient assignment of orders. The use of batch processing for orders and decision epochs helped in optimizing delivery routes and resource allocation. Increasing the number of dedicated vehicles at each restaurant under huge customer demand shows a slight increment in the order fulfillment rate, but led to a rise in overall cost which indicates that the system performed better when occasional drivers were used for the deliveries.

There are several directions for future research. One research direction is to incorporate a machine-learning based model to forecast the demand. This is an interesting way that will help the system to predict the customer order pattern and peak hour demand. Secondly, the design of the best compensation scheme for occasional drivers. Future research could be done on formulating the payment for ODs, which considers hourly payment, waiting periods, and other incentive schemes to support crowdshipping delivery. Lastly, a more sophisticated heuristics approach could be developed to solve larger instances. This could involve exploring techniques that allow for more efficient management of large-scale systems with thousands of restaurants, drivers, and customers. Advancements in technology and predictive modeling will benefit the transfer deliveries in crowdshipping environment.

References

[1] AppMySite, "70+ online food ordering statistics that every restaurateur should know," https://www.appmysite.com/blog/online-food-ordering-statistics/, 2024.

[2] Statista, "Online food delivery - worldwide," https://www.statista.com/outlook/dmo/online-food-delivery/worldwide?currency=usd/, 2024.

[3] McKinsey, "Ordering in: The rapid evolution of food delivery," https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/ordering-in-the-rapid-evolution-of-food-delivery/, 2024.

[4] J.-F. Rougès and B. Montreuil, "Crowdsourcing delivery: New interconnected business models to reinvent delivery," in *1st international physical internet conference*, vol. 1. Québec City (Canada) IPIC, 2014, pp. 1–19.

[5] C. Archetti, M. Savelsbergh, and M. G. Speranza, "The vehicle routing problem with occasional drivers," *European Journal of Operational Research*, vol. 254, no. 2, pp. 472–480, 2016.

[6] L. Dahle, H. Andersson, M. Christiansen, and M. G. Speranza, "The pickup and delivery problem with time windows and occasional drivers," *Computers & Operations Research*, vol. 109, pp. 122–133, 2019.

[7] L. Zhen, R. Baldacci, Z. Tan, S. Wang, and J. Lyu, "Scheduling heterogeneous delivery tasks on a mixed logistics platform," *European Journal of Operational Research*, vol. 298, no. 2, pp. 680–698, 2022.

[8] J. Parappathodi and C. Archetti, "Crowdsourced humanitarian relief vehicle routing problem," *Computers & Operations Research*, vol. 148, p. 105963, 2022.

[9] A. M. Arslan, N. Agatz, L. Kroon, and R. Zuidwijk, "Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers," *Transportation Science*, vol. 53, no. 1, pp. 222–235, 2019.

[10] H. Zhang, K. Luo, Y. Xu, Y. Xu, and W. Tong, "Online crowdsourced truck delivery using historical information," *European Journal of Operational Research*, vol. 301, no. 2, pp. 486–501, 2022.

[11] S. Fatehi and M. R. Wagner, "Crowdsourcing last-mile deliveries," *Manufacturing & Service Operations Management*, vol. 24, no. 2, pp. 791–809, 2022.

[12] F. Torres, M. Gendreau, and W. Rei, "Crowdshipping: An open vrp variant with stochastic destinations," *Transportation Research Part C: Emerging Technologies*, vol. 140, p. 103677, 2022.

[13] M. Silva, J. P. Pedroso, and A. Viana, "Deep reinforcement learning for stochastic last-mile delivery with crowdshipping," *EURO Journal on Transportation and Logistics*, vol. 12, p. 100105, 2023.

[14] J. Tao, H. Dai, W. Chen, and H. Jiang, "The value of personalized dispatch in o2o on-demand delivery services," *European Journal of Operational Research*, vol. 304, no. 3, pp. 1022–1035, 2023.

[15] M. Silva, J. P. Pedroso, and A. Viana, "Stochastic crowd shipping last-mile delivery with correlated marginals and probabilistic constraints," *European Journal of Operational Research*, vol. 307, no. 1, pp. 249–265, 2023.

[16] M. D. Simoni and M. Winkenbach, "Crowdsourced on-demand food delivery: An order batching and assignment algorithm," *Transportation Research Part C: Emerging Technologies*, vol. 149, p. 104055, 2023.

[17] S. Mancini and M. Gansterer, "Bundle generation for last-mile delivery with occasional drivers," *Omega*, vol. 108, p. 102582, 2022.

[18] I. Dayarian and M. Savelsbergh, "Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders," *Production and Operations Management*, vol. 29, no. 9, pp. 2153–2174, 2020.

[19] Y. Tao, H. Zhuo, and X. Lai, "The pickup and delivery problem with multiple depots and dynamic occasional drivers in crowdshipping delivery," *Computers & Industrial Engineering*, vol. 182, p. 109440, 2023.

[20] H. Bae and I. Moon, "Multi-depot vehicle routing problem with time windows considering delivery and installation vehicles," *Applied Mathematical Modelling*, vol. 40, no. 13-14, pp. 6536–6549, 2016.

[21] R. W. Calvo, "A new heuristic for the traveling salesman problem with time windows," *Transportation Science*, vol. 34, no. 1, pp. 113–124, 2000.

[22] G. Macrina, L. D. P. Pugliese, F. Guerriero, and G. Laporte, "Crowd-shipping with time windows and transshipment nodes," *Computers & Operations Research*, vol. 113, p. 104806, 2020.

Acknowledgments

I would like to express my heartfelt gratitude to Dr.Roberto Roberti for his invaluable guidance, support, and encouragement throughout this research. I am truly thankful for his supervision.

A
Appendix

The output of the delivery model using only direct delivery by occasional drivers is shown in table A.1.

Restaurants	Occasional Drivers	Customers	Total Cost (in euros)	Order Fulfilment (%)	On-time Arrival (%)	Orders sent to Pending (%)
5	20	25	3080.92	84.0	95.23	16.0
5	40	50	3609.73	50.0	76.0	50.0
5	80	100	5645.11	52.0	71.15	48.0
10	20	25	3408.29	52.0	100.0	48.0
10	40	50	5441.09	66.0	96.96	34.0
10	80	100	5893.05	52.0	86.53	48.0

Table A.1: Results for Delivery with Occasional Drivers (Only Direct Delivery)