

ESSEC
BUSINESS SCHOOL

Optimizing Growth: Decision Models for Last-Mile Delivery and Startup Scaling

A DISSERTATION PRESENTED
BY
Yuanyuan Li

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy in Business Administration
AND
Doctor en Sciences de Gestion
de l'Ecole Doctorale
«Economie, Management, Mathématiques, Physique et Sciences Informatiques»
ED 405
CY Cergy Paris Université

IN THE SUBJECT OF
Operations Management and Operations Research

Presented and publicly defended on the 9th of July, 2025

Jury

Claudia Archetti	Co-supervisor	Professor, University of Brescia (Italy)
Ivana Ljubić	Co-supervisor	Professor, ESSEC Business School (France)
Niels Agatz	Referee	Professor, Erasmus University Rotterdam (Netherlands)
Tom Van Woensel	Referee	Professor, Eindhoven University of Technology (Netherlands)
Felix Papier	President	Professor, ESSEC Business School (France)
Guillaume Roels	Examiner	Professor, INSEAD (France)

©2025 – Yuanyuan Li
all rights reserved.

Optimizing Growth: Decision Models for Last-Mile Delivery and Startup Scaling

Abstract

Both large-scale e-commerce firms and small startups face unprecedented opportunities for growth, yet they must overcome significant operational challenges to scale efficiently. This dissertation focuses on two application domains—last-mile delivery operations and startup growth. It explores how operational decision models can be used to optimize growth strategies and guide resource allocation in fast-growing and competitive environments.

Chapter 2 presents a structured review of emerging optimization problems for distribution in same-day delivery logistics. It first categorizes existing research, maps the delivery process, identifies sources of uncertainty, and reviews solution approaches. Then it compares solution structures obtained by applying the different formulations and discusses the corresponding implications. This chapter also provides managerial insights and outlines directions for future research. A paper based on this work has been accepted for publication in Computers and Operations Research.

Chapter 3 studies the orienteering problem with stochastic and dynamic release dates in same-day delivery. Two reinforcement learning approaches based on value function approximation are proposed. One combines deterministic integer linear programs with a consensus function, while the other incorporates a two-stage stochastic program. The chapter shows that exact methods can be competitive in real-time routing environments. A paper based on this work is published in Transportation Science. The work was presented at the EURO 2022 in Helsinki,

Finland, the POMS China 2023 in Zhejiang, China, and the TSL 2024 workshop in Nantes, France.

Chapter 4 explores a delivery model that uses autonomous ground vehicles to transport parcels for direct deliveries from a central depot to a pickup station in each trip. This chapter analyzes the static and deterministic version of the problem and its variants, derives performance bounds, and evaluates the proposed heuristics through worst-case and numerical analyses. A paper based on this chapter is in progress.

Chapter 5 investigates how startups can manage growth by balancing the costs of capacity adjustment with the need to reach a target market size. The analysis reveals that entrepreneurs should follow a strategic growth trajectory rather than pursuing aggressive expansion. In stable markets, increasing sales effort gradually is optimal, while in turbulent conditions, a more cautious and delayed ramp-up is preferred. A paper based on this chapter is in preparation. This work was presented at the POMS 2024 conference in Minneapolis, US, and the INFORMS Annual Meeting 2024 in Seattle, US.

Chapter 6 concludes the dissertation by summarizing the major findings from each chapter and outlining opportunities for future research.

Contents

1 Introduction **1**

1.1 Growth in dynamic markets 2

1.2 Growth challenges in last-mile delivery 3

1.3 Scaling up in startups 5

1.4 Decision models for optimizing growth 7

1.5 Overview of the dissertation 12

2 Emerging Optimization Problems for Distribution in Same-day Delivery **15**

2.1 Introduction 16

2.2 Overview of SDD problems 19

2.3 Release dates and deadlines 27

2.4 Time window options for attended home delivery 41

2.5 Delivery options 48

2.6 Pickup and delivery 62

2.7 Districting - fleet size and mix 68

2.8 Model comparison 73

2.9 Managerial insights and future research directions 79
2.10 Conclusion 84

3 Reinforcement Learning Approaches for the Orienteering Problem with Stochastic and Dynamic Release Dates 86

3.1 Introduction 87
3.2 Literature review 90
3.3 Problem description 96
3.4 Upper bounds and partial characterization of optimal policy 104
3.5 Reinforcement learning solution approaches 109
3.6 Computational experiments 126
3.7 Conclusion 153

4 Delivery with Pickup Stations and Autonomous Ground Vehicles 155

4.1 Introduction 156
4.2 Model 159
4.3 Complexity and heuristics for problem variants 162
4.4 The original problem: multiple station options per parcel and capacitated pickup stations (DOP) 170
4.5 Numerical experiments 175
4.6 Conclusion 184

5 Managing startup Growth Before Facility Investments 186

5.1 Introduction 187
5.2 Literature review 191
5.3 Model 194
5.4 Analysis of the deterministic formulation 199

5.5 Extension to a stochastic setting 210
5.6 Numerical experiments 211
5.7 Conclusions 217

6 Conclusion and Future Directions **220**

6.1 Summary of contributions 221
6.2 Limitations 222
6.3 Future research directions 223

Appendix A Appendix **226**

A.1 Appendix-Chapter 4 227

References **248**

Listing of figures

2.1 Classification scheme of SDD problems surveyed in this article 18

2.2 SDD stakeholders 19

2.3 SDD process with sources of uncertainty 20

2.4 Classification scheme describing characteristics of SDD problems 28

2.5 Small instance and corresponding solutions. The starting time of each trip in each block from left to right in order: 16, 46 and 130; 0, 53, 83 and 123; 28, 80, 144 and 184; 28, 80 and 144 . . 75

2.6 New instance and solutions obtained after changing the WTP and release dates of orders 5 and 8. The starting time of each trip in order: 32 and 138; 60 and 150 77

2.7 New instance and solutions obtained when changing the radius for determining feasible pickup stations. The starting time of each trip in each block from left to right in order: 31 and 89; 0, 46 and 126; 0, 46 and 130 78

3.1 The assignment of requests to batches (*a dashed line with no arrow connects known requests with batches from K_0, indicating potential assignments of each $i \in N_e^{known}$ to some $k \in K_0$)* 116

3.2 Performance of the five approaches based on the values of β (dispersion of release dates) 139

3.3 Performance of the five approaches based on the values of δ (rate of customers with dynamically updated release dates) 140
3.4 Performance of the five approaches based on the value of c 144
3.5 Left: VFA-CF returns five routes serving 34 requests, starting at time 0, 56, 149, 261, and 390. Right: VFA-2S returns five routes serving 36 requests, starting at time 0, 56, 143, 276, and 381. . 148
3.6 Left: MH returns three routes serving 27 requests, starting at time 0, 56, and 242. Right: ME returns four routes serving 31 requests, starting at time 0, 56, 236, and 469. 149
3.7 OPrd_PE returns six routes serving 31 requests, starting at time 0, 56, 118, 152, 184, and 329 . . 149

4.1 Location plot for an instance with 10 stations and 250 customers in Dataset 1. 177
4.2 Taken from Ulmer & Streng (2019) 178
4.3 Comparison of average served (%) and running time for different values of h when the time horizon is 720 and radius is 0, 10, and 20. 179

5.1 Illustration of total costs incurred in the two cases, assuming the startup reaches k within N periods, with parameters $k = 100, m = -5, q = 1$, and $c = 2$. The star indicates the optimal cost when $N = 1$ and failing to reach k is allowed with a penalty, given by $g_N(x_N) = p\,k - p\,x_N$, where $p = 107$. 200
5.2 The changes of growth trajectories for different target k ($m = -10, q = 3, c = 3$). 205
5.3 Optimal cost increasing in target recurring sales k (other parameters $m = -10, q = 3$, and $c = 3$). 207
5.4 Plot of sales effort over time ($a = 1.7$). 213
5.5 Plot of sales effort over time with different convexity levels of capacity adjustment costs. 215
5.6 Plot of sales effort over time with different variances in the same graph 216
5.7 Plot of accumulated recurring sales over time with different variances in the same graph 217
5.8 Plot of sales effort over time with different convexity levels of capacity adjustment costs 218
5.9 Accumulated recurring sales over time with different convexity levels of capacity adjustment costs 218

This is dedicated to my mom—who, used all her strength, lifted me high enough to see a broader sky.

Acknowledgments

Writing this dissertation has been a rewarding but challenging journey. I could not have completed it without the support of many people, to whom I am deeply grateful.

First and foremost, I am grateful that I had two amazing supervisors, Prof Claudia Archetti and Prof Ivana Ljubić. I could never have pursued or developed this research without the knowledge and guidance they taught me. Their passion for research and pursuit of excellence have been a constant source of inspiration to me. I have told myself many times that being even half of what they are would already be an achievement. Claudia, thank you especially for your kindness and support throughout my PhD journey across France and Singapore. Ivana, thank you always taking the time to listen to my doubts and explain my questions with clarity and care. Your mentorship has meant a lot to me.

I am also deeply grateful to Prof Felix Papier. I have always admired your kindness, patience, and remarkable expertise across so many areas. You were the first to teach me how to deliver a professional academic presentation, and you provided unwavering support throughout my job search. I feel truly fortunate to work with you and to explore the startup research journey together. I cannot fully express how thankful I am for your guidance and encouragement.

I am lucky to have had the opportunity to work with Prof Geoffrey Chua, whose research passion, clarity of thought have always impressed me. I also wish to sincerely thank my coauthor, Prof Riccardo Mogre, who has always been willing to address every question I brought forward. I would also like to thank my master's advisor, Prof Roberto Tadei, and online mentor Prof Jim Bookbinder for their generous support. Despite their busy schedules and seniority, they have always been willing to help young researchers like me by sharing their valuable advice.

I thank Prof Niels Agatz, Prof Tom Van Woensel, and Prof Guillaume Roels for agreeing to serve on my PhD committee and for offering valuable feedback on my work. It has been an honor to have you involved in this process.

I am grateful to have been part of the ESSEC tribe. Prof. Laurent Alfandari was the first professor to warmly welcome us. His availability to students, excellent teaching materials, and professionalism have always impressed me. I also want to thank the PhD office—Christine, Audrey, and Iliona, who are always available for us, handling all the administrative matters so efficiently and ensuring our PhD journeys were as smooth as possible. Throughout this journey, I've been fortunate to be surrounded by a supportive group of fellow students and friends. I would like to thank Minakshi, Yanlu, Alborz, Java, Deba, Yerlan, Patrick, and Baolong for the valuable advice and support you have shared with me. I'm also grateful to Wenjin, Hui, Shadi, Hao, Xin, Xiaowei, and Coco for being there through all the ups and downs. Your presence has meant a lot.

Finally, I thank my mom for her unconditional support and unwavering belief in me. I am here because of you. I am also deeply grateful to my boyfriend, Tianhao. Your patience, love, and all-encompassing support helped me through the toughest times during the PhD journey. Thank you, from the bottom of my heart.

1

Introduction

1.1 Growth in dynamic markets

In today's economy, both large-scale e-commerce firms and small startups face unprecedented growth opportunities. According to recent estimates, global retail e-commerce sales reached an estimated 5.8 trillion U.S. dollars in 2023 and are projected to surpass 8 trillion dollars by 2027, reflecting a 39 percent growth over the coming years [1]. At the same time, entrepreneurial activity has surged. Analysis from the Center for American Progress indicates that entrepreneurs in the United States submitted 450,000 likely employer applications[2] in the fourth quarter of 2023, recording a 37% increase compared to the same period in 2019. These trends signal a significant growth in entrepreneurial activity and the establishment of new startups.

Yet, achieving growth in today's dynamic markets is not merely a matter of increasing capacity or attracting additional customers. It requires making a series of timely and strategic decisions that can adapt to uncertain, competitive, and evolving markets. As companies scale, their operations typically become more complex and interdependent. As a result, even routine operational decisions become more critical, since small mistakes can escalate and lead to larger disruptions. For large e-commerce platforms, such as Amazon, JD.com (JINGDONG), and Uber Eats, growth brings significant operational complexity, particularly in last-mile delivery. These firms must manage hypergrowth from day-to-day operations, ensuring that their logistics operations remain both cost-efficient and responsive, as customer expectations for speed and reliability continue to rise.

Startups, by contrast, face different but equally pressing challenges. Unlike established firms, they typically operate under severe constraints in capital, manpower, and infrastructure. Early decisions, such as when and how to invest in hiring, product development, or new technologies, can critically influence whether a startup achieves

[1] https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/

[2] https://www.americanprogress.org/article/entrepreneurship-startups-and-business-formation-are-booming-across-the-u-s/, likely employer applications are a subset of business applications that show a high probability of turning into businesses with payrolls.

sustained growth or faces premature failure.

This dissertation investigates how decision models can inform and optimize growth strategies across these two settings. Specifically, it focuses on last-mile delivery operations in the context of high-growth e-commerce delivery companies and the scaling up of startups. In this context, operational decision models refer to optimization frameworks that guide resource allocation and timing decisions in fast-growing environments, which can serve as the basis for analyzing and supporting growth-related decisions in both application domains. In the following, I provide further background and motivation for each of these two applications.

1.2 Growth challenges in last-mile delivery

As global e-commerce sales are growing, customer expectations for service quality are also rising. A recent survey shows that almost half (48%) of shoppers prioritized delivery speed, 43% seek reduction of the cost of delivery, and 39% value more reliable delivery time information (Statista, 2025). These expectations are reshaping how e-commerce platforms structure their delivery operations. For instance, Cui et al. (2020) highlight the tension between SF Express and Alibaba, showing that providing reliable and timely delivery, especially for expensive or less-discounted products, brings better customer experience and thus higher profits.

One of the most critical components affecting both service quality and cost in e-commerce logistics is the last-mile delivery leg (Archetti & Bertazzi, 2021). Last-mile delivery refers to the final segment of a product's journey from a distribution hub to a final destination (e.g., customer's home). While freight networks efficiently move parcels across long distances using trains, ships, or trucks, the last leg requires deliveries to be made in smaller batches to dispersed destinations. This segment is typically the least efficient and most expensive part of the delivery process due to factors such as limited consolidation opportunities, customer expectations for speed, and urban congestion.

To stay competitive, many e-commerce firms are investing in same-day delivery services. While these services offer greater convenience to consumers, they also create substantial operational challenges for delivery companies. Since customers expect deliveries to be completed within hours of ordering, companies often need to begin dispatching and routing operations before all parcels have arrived at the depot. The processes of order arrival, acceptance, picking and packing, and delivery must all occur within the same day, forcing firms to rethink warehouse layouts, routing algorithms, and workforce scheduling.

In response to these pressures, companies are increasingly turning to technology to enhance last-mile operations. For example, Waymo, an American autonomous driving technology company, collaborates with Uber Eats to deliver food (Waymo, 2024). Similarly, Zipline, an American drone delivery company, has achieved significant milestones, completing over one million commercial drone deliveries (Zipline, 2024). Meanwhile, IoT-enabled devices are being used for tracking and securing deliveries, as adopted by DHL (DHL, 2024). Besides, to reduce failed deliveries and improve flexibility, many platforms now offer customers delivery location options, such as home drop-off or pickup stations. While these technological innovations offer greater flexibility and service quality, they also introduce new layers of operational complexity. To understand the challenges of managing last-mile delivery—particularly in same-day settings—it is helpful to examine the underlying workflow.

In a typical same-day delivery process, once a customer places an order, the company must quickly negotiate delivery terms—such as time windows, locations, and prices—based on current delivery capacity. If an agreement is reached, the order is accepted and moves to the dispatching stage, where the company decides when and how to send out the order. Finally, vehicle routing decisions are made, often under time pressure and with limited information about future orders. These decisions—order negotiation, dispatching, and routing—are tightly linked and must be made repeatedly throughout the day.

Each of these delivery steps is subject to operational uncertainty. Customer preferences are not known in advance when offering customers a variety of delivery options, such as different time slots, delivery methods, or

locations. The timing of when customer orders become ready at the depot (often referred to as release dates in the literature) is also uncertain. In meal delivery settings, order preparation times vary significantly across restaurants. In crowdsourced logistics systems, driver availability may fluctuate. External factors, such as traffic congestion, can further disrupt planned routes.

When companies fail to manage their delivery operations effectively, their growth trajectories can be disrupted. For instance, Amazon's drone delivery service, Prime Air, faced setbacks due to operational challenges. In January 2025, the company suspended its commercial drone deliveries in Texas and Arizona after two drones crashed during rainy weather[3]. Another example is with the UK's Royal Mail[4]. In December 2024, the company was fined £10.5 million by the Office of Communications for failing to meet its delivery target—delivering only 74.7% of mail within one working day against a required 93%, and its underperformance not only led to financial consequences but also hurt public trust.

Thus, it is critical to manage delivery operations effectively. Achieving scalable and profitable growth in last-mile delivery in an environment full of uncertainties often requires the use of advanced decision models capable of sequential decision-making under uncertainty. These models support real-time operational adjustments that align with long-term strategic goals and enable firms to better cope with dynamic customer demands and external disruptions. The decision models will be discussed in Section 1.4.

1.3 Scaling up in startups

The entrepreneurial life cycle can be divided into three stages: nailing, scaling, and sailing (Fine et al., 2022). In the nailing stage, a startup focuses on defining its value proposition—developing a viable product or service, under-

[3]https://www.wired.com/story/texas-amazon-drones-stop-flying/

[4]https://www.ft.com/content/765f3fa9-1bbe-4985-b536-9a5ac613d6e4

standing customer needs, choosing technologies, and experimenting with pricing strategies. Once key aspects of the value proposition have been proven, the startup enters the scaling stage, where the priority shifts to growing its market along with its production and delivery capabilities. Finally, the sailing stage comes after the company has captured a considerable share of its target market. At this stage, growth tends to slow, and the business prioritizes business sustainability, operational efficiency, and incremental improvement.

Each of these stages presents unique challenges. The scaling stage is particularly critical: it is the point at which the startup must transition from early success to sustainable growth. However, unlike mature firms, startups often lack the operational infrastructure, managerial experience, and financial buffer to support production expansion. Key systems, such as hiring pipelines, supply chains, customer service, and delivery logistics, are often still in development, making the scaling process both complex and risky.

This challenge is especially relevant in the current environment. We are in the midst of a startup boom, yet high setup failure rates persist. Research by Eisenmann (2021) indicates that most startups fail to succeed, with more than two-thirds never delivering a positive return to investors. In a survey of 500 startup owners, nearly half identified insufficient funding as the primary reason for their business failure[5]. These findings show the critical role of financial resources in enabling startups to survive and scale. Expansion often requires substantial investment in infrastructure to meet the production and delivery capacity. To attract such funding, startups must demonstrate strong growth potential, often through building a strong and growing customer base (Soufani, 2001).

However, growth itself can become a source of risk. As Eisenmann (2021) observes, fast growth may tempt founders to shortcut essential processes such as comprehensive customer research and product iteration. It can also put heavy demands on members and partners, leading to quality control problems, supply chain disruptions, or declining customer satisfaction. These risks can erode profit margins and damage long-term viability.

[5]https://www.statista.com/statistics/1271464/start-up-failure-reasons/

Examples from practice illustrate these dynamics. SkinnyGirl Cocktails faced operational setbacks during periods of rapid growth. Their initial collaboration partner struggled to meet exploding demands, resulting in being sold to a larger company at a value that appears below its future cash flows (Fine et al., 2022). Similarly, Banza, a startup that produces healthy and gluten-free chickpea pasta, struggled to replicate its home kitchen recipe on a large scale. These results lead to customer dissatisfaction. Only after redirecting its focus from marketing to improving production processes did the company achieve more stable growth and successful expansion (Fine et al., 2022).

A sudden surge in orders can lead to high capacity adjustment costs, including expenses related to hiring and training new workers, managing production disruptions, and addressing quality-related rework. We see the importance of managing the growth process at an appropriate pace while accounting for these adjustment costs. Ultimately, startups must strike a careful balance: growing quickly enough to capture market opportunities and secure investor interest, while avoiding the operational risks that often accompany premature or poorly managed growth. Making optimal decisions about when and how to grow is a fundamental challenge in the scaling stage—one of the core focuses of this dissertation.

1.4 Decision models for optimizing growth

In both last-mile delivery and startup scaling, managing growth effectively often requires making sequential operational decisions under uncertainty. These decisions must account for demand fluctuations, resource limitations, or cost trade-offs. This chapter presents a modeling framework that captures such dynamics through the lens of sequential decision-making. I first introduce the common modeling framework and solution approaches for dynamic and stochastic problems in Section 1.4.1, then, a comparative discussion of the modeling choices is provided in Section 1.4.2.

1.4.1 Sequential Decision-Making for Growth Management

To model dynamic systems under uncertainty, researchers often adopt the framework of Markov Decision Processes (MDPs) (Puterman, 2014). MDPs provide a structured approach for modeling decisions made over time under uncertainty. The components of a sequential decision-making process are defined as follows (Powell, 2014):

- State S_t: It represents all the information needed to make a decision at time t.
- Actions/Decisions x_t: They depend on the nature of the problem. The decision is taken according to a decision rule $X^\pi(S_t)$ that maps a state S_t to a feasible action x_t. A solution of a sequential decision-making process is a policy π that is a sequence of decision rules $\pi = (X^\pi(S_0), ..., X^\pi(S_T))$ mapping states to actions, where $x_t = X^\pi(S_t)$.
- Exogenous information W_{t+1}: It is the new information available during the tth iteration after making decision x_t. This means that states, actions/decisions, and information evolve as follows:

$$(S_0, x_0, W_1, S_1, x_1, W_2, ..., S_t, x_t, W_{t+1}, ..., S_T). \tag{1.1}$$

- Transition function $\mathcal{S}$: It consists of the equations that transit the system from t to $t+1$:

$$S_{t+1} = \mathcal{S}(S_t, x_t, W_{t+1}). \tag{1.2}$$

- Objective function: It specifies the costs being minimized, the rewards being maximized, or other performance metrics. For example, let $C(S_t, x_t)$ be the cost we are minimizing given the decision x_t and state S_t. The goal is to find a policy that minimizes expected costs over the entire time horizon, which reads as

follows:

$$\min_{\pi} \mathbb{E}\left(\sum_{t=0}^{T} C(S_t, X^{\pi}(S_t))|S_0\right), \tag{1.3}$$

where $\min_{\pi}$ indicates we are finding the best policy, and we take the expectation $\mathbb{E}$ of the sum of costs, which averages over all the possible outcomes of the information process $W_1, \ldots, W_T$ that arrive over time.

The solution approaches to derive policies for a sequential decision-making process depend on the exploitation of the quality and the evolution of information. Based on the methods described in Soeffker et al. (2022), we discuss the following solution approaches: value function approximation (VFA) approaches, scenario-based approaches, rollout algorithms, rolling horizon heuristics, policy function approximation (PFA) or cost function approximation (CFA) approaches, and more.

VFA, scenario-based approaches, and rollout algorithms integrate sampled information realizations into the decision-making processes.

VFA estimates the value of being in a particular state S_{t+1} after taking an action x_t in state S_t. It approximates the future impact of decision x_t. In Operations Research, this is commonly referred to as VFA, while in Computer Science, it is often studied under the term Q-learning. Specifically, Q-learning estimates values of state-action pairs, known as Q-factors, denoted as $Q(S_t, x_t)$. A Q-factor incorporates the immediate cost $C(S_t, x_t)$ and the future costs. The best decision can be found by calculating $Q(S_t, x_t)$ for each S_t and x_t, i.e., solving $min_x Q(S_t, x_t)$. Since state spaces are often large, $Q(S_t, x_t)$ is often approximated using various approaches, such as neural networks.

In scenario-based approaches, a set of static and deterministic scenarios is generated by sampling the information. These scenarios can be solved individually using mixed-integer solvers or heuristics, or simultaneously with multi-stage decision problems (Dupačová et al., 2000). One notable scenario-based approach is the multiple-scenario approach proposed by Bent & Van Hentenryck (2004), which continuously generates solutions, selecting a given solution via a consensus function.

Rollout algorithms, first introduced by Bertsekas et al. (1997), involve preselecting a set of potential decisions and evaluating their possible outcomes by simulating actions within a given context. For example, in a chess game, a player can assess the quality of a move by simulating a sequence of moves once the initial decision is made.

Another type of approach, rolling horizon heuristics, has been used for decision-making in dynamic and stochastic environments for decades. This approach optimizes over an interval $(t, t+H)$, implementing decisions for time t, rolling to $t+1$, and repeating the process over the interval $(t+1, t+H+1)$. Both rolling horizon and rollout algorithms are look-ahead policies (the set of policies that model the future trajectory of each decision by optimizing over a restricted set of future periods). We refer to Sethi & Sorger (1991) for the theoretical framework of rolling horizon heuristics and Chand et al. (2002) for a summary of papers contributing to rolling horizon heuristics.

The following two method classes, PFA and CFA, make decisions based on a preliminary problem analysis, often using simplified assumptions about the information and the problem settings. PFA directly maps a state to an action (using, e.g., a parametric model) without solving any embedded optimization problem. CFA involves solving a deterministic optimization problem while modifying the constraints or the objective function. For example, in a resource allocation scenario, one might reduce the capacity of available resources in the constraints to maintain spare capacity for unexpected job arrivals; in meal delivery applications, a penalty cost can be introduced in the objective function to avoid late arrivals to mitigate uncertainty in food ready time (Ulmer et al., 2021). For further details on these two approaches, we refer to Powell (2021).

We refer to the survey by Soeffker et al. (2022) and the book by Powell (2021) for an exhaustive overview of sequential decision-making processes and the advantages and disadvantages of the various approaches. The methods described above can be hybridized to obtain a better strategy. For instance, one can first apply a rolling horizon method for a specified number of periods, followed by a VFA to approximate the states at the end of the specified periods (see Chapter 11.7 in Powell (2021)). Users can select their preferred solution methods based on the characteristics of the specific problem, the features of each approach, and the existing methods used in similar contexts,

as discussed in the following chapters.

Each of these methods offers trade-offs between model detail, computational tractability, and adaptability. In this dissertation, these approaches are tailored to the characteristics of each application area. For example, in last-mile delivery, real-time vehicle routing decisions are made using the hybrid approaches of VFA and scenario-based approaches (see Chapter 3).

1.4.2 Comparative Discussion and Method Selection

While both last-mile delivery and startup growth involve managing growth under constraints, the nature of decisions and uncertainty differ significantly. Delivery operations require real-time responsiveness, making numerical and heuristic methods introduced in Section 1.4.1 more suitable. In contrast, startup scaling involves more strategic, long-term planning, where decisions unfold over extended time horizons.

In this dissertation, startup growth is modeled using a deterministic dynamic optimization framework, leading to analytical results that characterize the structure of optimal policies. These results provide managerial insights into the pacing of growth.

To incorporate uncertainty, the model is then extended to a stochastic setting in which the number of new sales resulting from sales effort is uncertain, reflecting the market uncertainty. Due to the complexity introduced by this randomness, we first rely on numerical simulations to gain insights and to illustrate how risk and uncertainty influence optimal strategies, leaving analytical explorations for future work.

The modeling approaches in this dissertation are thus selected based on the structure and timing of decisions in each application domain. Together, they demonstrate how operational decision models can be adapted to different forms of growth, offering both day-to-day decision-making and long-term strategy in different domains.

A more detailed discussion of model specifications and solution methods is presented in the subsequent chapters for each application.

1.5 Overview of the dissertation

This dissertation explores how operational decision models can support growth-related decision-making in two high-growth contexts: last-mile delivery and startup scaling.

In last-mile delivery, despite an extensive and growing body of literature prior to this work, several important research gaps remained:

- There is a range of emerging optimization problems in distribution in the context of same-day last-mile delivery. However, there is no comprehensive study that systematically categorizes these problems, identifies managerial insights, and outlines future research directions.
- Most existing work on stochastic and dynamic vehicle routing problems relies on heuristic policies, due to computational complexity and the strict real-time requirements. It remains an open question whether exact methods can play a greater role in solving these problems effectively.
- Many solution approaches are evaluated primarily based on average performance or theoretical guarantees. However, little attention has been paid to the possibility of selecting algorithms based on specific instance characteristics, which could improve real-world applicability and flexibility for various scenarios.

In the context of startups, existing research has largely focused on capacity expansion for mature firms or investment decisions around production, timing, and hiring. In contrast, the roles of sales effort—a critical factor in early-stage growth and securing funding—have received limited attention. This dissertation addresses this gap

by explicitly modeling the trade-offs involved in imposing sales effort to support growth under different market conditions.

To address these research gaps, this dissertation is structured as follows:

Chapter 2 develops a structured review of emerging optimization problems for distribution in same-day delivery. It categorizes existing research, maps the delivery process, identifies uncertainty sources and solution approaches, compares solution structures derived from different formulations, and provides managerial insights along with future research directions.

Chapter 3 studies the orienteering problem with stochastic and dynamic release dates (i.e., the moments when parcels are available at the depot) in same-day delivery. To address this sequential decision-making problem, two reinforcement learning approaches based on VFA are proposed. One combines deterministic integer linear programs with a consensus function, while the second integrates a two-stage stochastic integer linear program. The chapter demonstrates that exact methods can offer meaningful benefits even in complex, real-time routing environments.

Chapter 4 explores a system that uses autonomous ground vehicles for direct deliveries from a central depot to pickup stations, aligning with the current autonomous vehicle capabilities by avoiding complicated routing difficulties. The chapter studies the static and deterministic version of the problem and analyzes some of its variants by studying its performance bounds, performing worst-case analyses, and comparing the proposed heuristics numerically. The current version is a work-in-progress. Although the initial scope intended to cover dynamic parcel arrivals, the current version focuses on the simpler case and is left open for future extension.

Chapter 5 investigates how startups should manage growth by making the tradeoff between the growth pains and risks represented with capacity adjustment costs and the need to reach a target market size. The findings reveal that entrepreneurs should follow a strategic growth trajectory rather than pursuing aggressive expansion. In stable markets, gradually increasing sales efforts is optimal, while in a turbulent market environment, entrepreneurs are

suggested to adopt a more cautious approach by delaying growth speedup.

2

Emerging Optimization Problems for Distribution in Same-day Delivery

joint work with **Claudia Archetti** and **Ivana Ljubić**

2.1 Introduction

Same-day delivery (SDD) has increasingly become a vital strategy for retail businesses aiming to stay competitive (Roadie, 2024). It ensures that orders placed online are delivered within the same calendar day. Current estimates suggest that the same-day delivery market in the United States will grow from 9.25 billion U.S. dollars in 2024 to approximately 13.15 billion U.S. dollars in 2030 (Placek, 2024). For example, GNC, a leading global health and wellness retailer in the US, faced increasing competition from e-commerce giants and rising consumer expectations for faster delivery. In response, GNC recognized the need to invest in its delivery capabilities to maintain its market position. In June 2024, the company partnered with Uber Eats to offer same-day delivery for online orders (GNC, 2024). Progress in technology has played a key role in facilitating faster and more convenient delivery options, with autonomous vehicles and drones increasingly being used for shipping.

Although SDD services bring advantages to consumers, they also create challenges for SDD companies, as the arrival of requests, their acceptance, picking and packing, and delivery to customer locations must all occur within the same day. This involves restructuring delivery systems, ranging from warehouse design to delivery processes. Recent surveys by Bock (2021), Archetti & Bertazzi (2021) and Boysen et al. (2021) briefly review some strategies in online routing designed for urgent delivery processes and cover a broad range of relevant and emerging optimization problems in e-commerce and last-mile delivery. However, they do not specifically focus on SDD and its corresponding optimization challenges. Meanwhile, distribution operations under uncertainty have been studied in the context of dynamic and stochastic vehicle routing, as seen in a comprehensive survey by Zhang & Van Woensel (2023). The survey mainly focuses on the uncertainty concerning random arrivals of customer requests and customer locations. In contrast, the sources of uncertainty in SDD are much richer and may also involve product arrival times at the depot, food ready times, vehicle/courier availability, and customer choices, as we will see later. Moreover, Zhang & Van Woensel (2023) consider SDD problems as a new sub-variant of the dynamic vehicle routing problem (VRP), so they focus on and refer to a restricted set of papers. Instead, we provide a

more comprehensive review of SDD works, present formulations for static problems, emphasize challenges, derive important managerial insights, and identify future research directions in SDD.

This survey aims to build a fundamental understanding of the emerging optimization problems for distribution in SDD by reviewing and categorizing existing research. Specifically, we focus on problems that embed new features related to the nature of SDD and which, as such, render the corresponding optimization problems different from problems faced in the former literature. Due to the limited space, we mainly focus on distribution operations and do not cover problems related to, e.g., inventory management. In SDD, order arrival, assembly, vehicle routing, and delivery happen within a single day. Since orders arrive during distributions, we focus on *order-based fulfillment systems*, where the SDD company decides dynamically on each customer request whether to start assembly and delivery of orders arrived up to that time, as defined by Waßmuth et al. (2023). Notice that these systems work differently than the *periodic fulfillment* ones, where a cut-off time is set to order arrivals, distribution happens afterward, and more flexibility is provided in dealing with dynamic order arrivals. This chapter focuses on the operational and tactical planning levels of SDD services (as shown in Figure 2.1). We acknowledge that the lists of operational and tactical problems presented herein may not be exhaustive, as the landscape of SDD problems continuously evolves. The following classes of relevant routing problems in SDD have been identified:

- Routing problems with release dates and deadlines (see Section 2.3);
- Time window options in attended home delivery services (see Section 2.4);
- Delivery options, including where to deliver and how to deliver with non-traditional transportation modes (see Section 2.5);
- Pickup and delivery problems (see Section 2.6);
- General design of SDD systems, involving districting problems, and extensions (see Section 2.7).

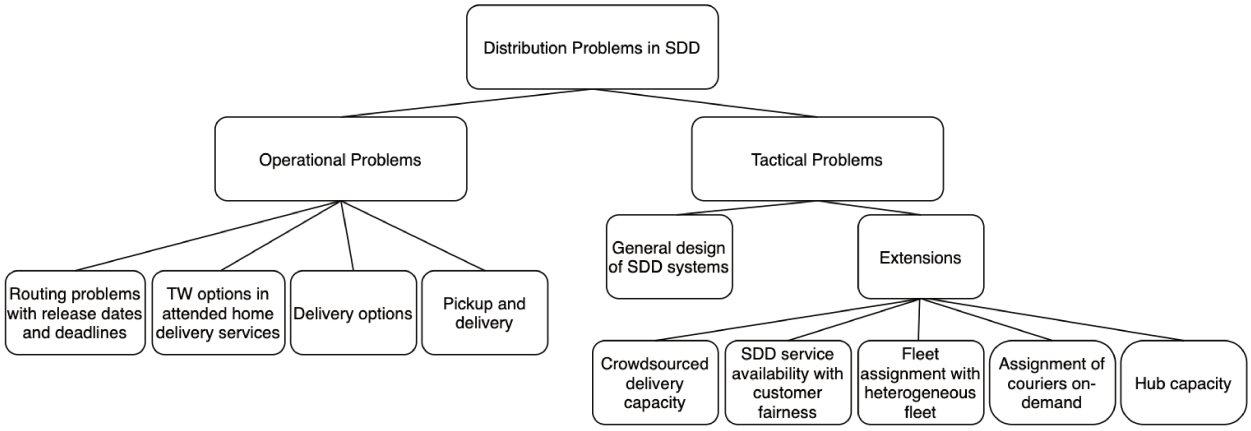


Figure 2.1: Classification scheme of SDD problems surveyed in this article

Among the five classes of problems presented in this work, the first four mainly deal with operational decisions; the last one (districting) primarily concerns the general design of SDD systems and extensions incorporating additional features, such as crowdsourced delivery capacity and customer fairness. In operational problems, problems with release dates and deadlines are the basis for many other SDD problems that often include additional complexities beyond these temporal constraints. Thus, we start our discussion in the subsequent sections by introducing the problems with release dates and deadlines.

We present mathematical formulations for routing problems with release dates, time window options, and delivery options with pickup stations. The models are built for basic settings, i.e., deterministic case and a single vehicle. These models help to understand each problem setting and can be used as a basis of solution approaches for more complex systems or to design *a priori* solutions (Klapp et al., 2018a), in reoptimization solution schemes (Archetti et al., 2020a; Pina-Pardo et al., 2024a), for competitive analysis in case of stochastic settings (Li et al., 2024b; Stroh et al., 2022), or as tools embedded in solution approaches for more general problems. A more detailed description of the use of deterministic models can be found in Section 2.2.1.2. For the other two classes of problems, i.e., pickup & delivery and districting, we do not present problem formulations as they can be obtained by combining the traditional models with Formulation (2.1), i.e., the first formulation presented in the current

work. Finally, we summarize managerial insights, identify major challenges, and specify future research directions.

The rest of the chapter is structured as follows. Section 2.2 provides an overview of SDD problems by structuring process flows, identifying the main sources of uncertainty, and describing approaches to handle the uncertainties. Subsequently, it presents classification schemes for distribution problems in SDD. The different classes of operational and tactical problems are reviewed in Sections 2.3-2.7. Section 2.8 compares the formulations and the corresponding solutions on small instances. Section 2.9 presents managerial insights and future research directions. Section 2.10 concludes the chapter.

2.2 Overview of SDD problems

In this section, we first highlight the primary stakeholders in SDD services, explain the main process flows, and discuss the various uncertainty factors in Section 2.2.1. Next, we present a classification scheme for SDD problems in Section 2.2.2.

2.2.1 Major stakeholders in SDD, process flows and sources of uncertainty

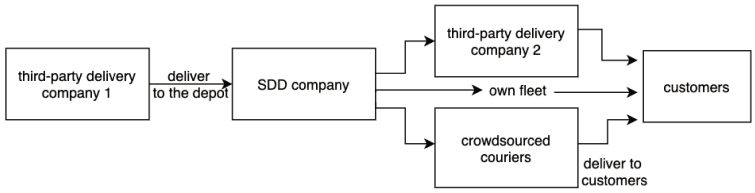


Figure 2.2: SDD stakeholders

SDD services manage day-to-day operations in an environment characterized by continuous customer order arrivals. The products being delivered through this system can be newspapers (NewspaperDirect, 2024), groceries, medicines, and fashion-related items, which are frequently requested through SDD services.

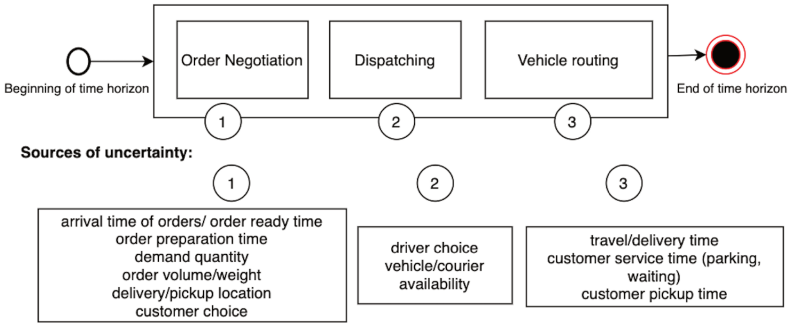


Figure 2.3: SDD process with sources of uncertainty

Key stakeholders include online retailers, customers, third-party delivery companies responsible for delivering parcels to depots and/or from depots to customers, and crowdsourced couriers who deliver using their vehicles/bicycles or on foot on a freelance or contract basis (Verlinde & Macharis, 2016; Archetti & Bertazzi, 2021; Fleckenstein et al., 2023). In Figure 2.2, we present the connections among the stakeholders.

2.2.1.1 Process Flows in SDD

In SDD, customer orders are received continuously throughout distribution operations, which makes it necessary to develop a rapid decision-making system regarding order negotiation, dispatching, and vehicle routing. *Order Negotiation* involves negotiating the delivery terms with a customer, ensuring the delivery is feasible and cost-effective. Meanwhile, *Dispatching* and *Vehicle Routing* focus on optimizing delivery routes and schedules based on available resources. The process iterates until the end of the time horizon or until no other orders can be accommodated in the ongoing distribution plan. Objectives may include optimizing various performance metrics, such as maximizing expected request fulfillment, maximizing profit, minimizing vehicle operating costs, or minimizing CO_2 emissions as part of environmental sustainability goals (Dobush, 2024). The process is summarized in Figure 2.3. Detailed decision specifications are provided below.

We define *Order Negotiation* as the agreement between a customer and an SDD company on when and where

to deliver. Initially, the company usually presents the customer with various options, which may include different deadlines, time windows, locations, and corresponding prices. The customer then selects their preferred option if interested. Once the SDD company confirms, the agreement is established. However, if no feasible options are available or if the customer does not find any of the provided options satisfactory, the order is either not placed or scheduled for the next day.

Dispatching involves determining delivery schedules to fulfill orders. It typically includes assigning orders to drivers and deciding whether to immediately dispatch all or some of the current orders or wait for additional orders for potential consolidation.

Vehicle Routing determines vehicle routes for dispatched orders, where various delivery means may be employed, such as trucks, drones, hybrid combinations of vehicles and drones, crowdsourced couriers, or others, depending on the resources available and the operational context.

These decisions can be made sequentially or in an integrated way, depending on the specific operational model employed. A sequential approach may be more practical in some contexts, such as food delivery services. For example, Liu et al. (2021) describe a business model motivated by the operations of a food delivery service, where orders are first captured and then assigned to drivers, with the drivers subsequently choosing their preferred routes. Conversely, other contexts may follow an integrated approach to increase efficiency and reduce myopic decision-making. For instance, Voccia et al. (2019) present a scheduling policy that integrates these decisions for each new order in the context of SDD. Truden et al. (2022) explore an international grocery retailer's online shopping service, where a new order triggers an update of the available schedule by jointly determining assignment and routing.

It is essential to distinguish SDD from conventional or next-day delivery scenarios. In SDD, orders arrive continuously, rendering it impractical to wait for the aggregation of all parcel arrivals before initiating deliveries. Consequently, vehicles are often required to undertake multiple routes to meet the time windows stipulated by customers or to adhere to the delivery deadline. Thus, for each dispatch, not all available parcels need to be delivered

immediately, even when time permits, as some can be dispatched later, along with incoming parcels that share close delivery locations, optimizing overall time and distance.

As illustrated in Figure 2.2, multiple stakeholders are involved in SDD routing problems, each with distinct concerns and objectives. The mixed-integer linear programs (MILPs) presented in the following sections pertain specifically to the decisions made by the SDD company, i.e., the SDD company decides which location options or time windows to offer to each customer.

2.2.1.2 Sources of uncertainty and handling approaches

Sources of uncertainty: SDD systems have to deal with various sources of uncertainty. In Figure 2.3, we show the stages within the SDD process where uncertainty arises, and we list the corresponding factors of uncertainty below the stages.

When negotiating orders, common uncertainty factors include arrival time, demand quantity, order volume/weight, and delivery/pickup location. When an SDD company offers customers various delivery options—such as different time slots, delivery methods, or locations—uncertainty arises regarding which options customers will choose. In meal delivery, order preparation times are often uncertain. In dispatching and routing decisions, there is uncertainty in occasional drivers' preferences on the requests they would like to serve, especially when they have the flexibility to choose. Besides, vehicle/courier availabilities, as well as customer service time due to parking and waiting for package handover, are frequently dynamic and stochastic. Furthermore, external factors, such as traffic congestion, may induce variations in vehicle travel time during routing processes. In scenarios where parcels are delivered to pickup stations, customer pickup times are uncertain, which matters because of the limited capacity of pickup stations.

Handling approaches: To deal with uncertainty, Poisson processes, Gaussian, and uniform distributions are commonly employed to model request arrival times (e.g., Ulmer et al. (2021); Stroh et al. (2022)), while uniform

distributions are frequently assumed for request locations (Ulmer, 2017). These distribution assumptions predominantly serve for generating scenarios in solution frameworks and empirical studies. However, uncertainty factors can be modeled with additional considerations to be more applicable to real-world scenarios. For example, one can define a time-varying order rate over the service day (Banerjee et al., 2022), or time correlation can be introduced in travel time (Pureza & Laporte, 2008). Also, there might be situations where almost all vehicles or couriers become unavailable due to a sudden surge in demand, such as during shopping festivals (Yang, 2020), or due to unexpected issues like severe weather conditions or operational disruptions. When modeling vehicle or courier availability, considering these rare but impactful events helps companies prepare effective risk management strategies.

In the papers we reviewed, a predominant focus is addressing dynamic and stochastic problems, often modeled as sequential decision-making processes. A Markov Decision Process (MDP) is a specific type of sequential decision-making process that assumes the Markov property, which means the system's future state depends only on its current state and the chosen action, making it memoryless to past events (Puterman, 2014).

For modeling approaches that capture the stochastic and dynamic elements in vehicle routing problems, we refer readers to Ulmer et al. (2020) that propose route-based MDPs and Fleckenstein et al. (2023) that provide a generalized, high-level model formulation. For the solution approaches to derive policies for a sequential decision-making process, we refer to the descriptions in Section 1.4.1.

Using deterministic models for sequential decision making: As mentioned above, in this article we provide several deterministic mathematical models. These models can be used for stochastic and dynamic problems in the following ways:

- Definition clarification: They help understand the deterministic setting from which the stochastic and dynamic version derives;

- Perfect information bounds: They provide either an upper bound for a maximization problem or a lower bound for a minimization problem to stochastic and dynamic versions for assessing their quality. This is a common approach in literature (Lan et al., 2024; Li et al., 2024b; Stroh et al., 2022).

  The two goals match Yildiz & Savelsbergh (2019a), who study a meal-delivery problem assuming perfect information about the order-arrival stream.

- Basis for solution development: Deterministic models can act as a foundation for developing solutions to dynamic and stochastic problems.

  - A priori solutions: Deterministic formulations can provide initial solutions, which can then be updated as new information becomes available. For example, in Klapp et al. (2018a), the authors derive a priori solutions from deterministic formulations and update them heuristically in real time as conditions change.

  - Reoptimization solution schemes: Deterministic formulations can also be used in reoptimization solution schemes, i.e., one can run a deterministic formulation with updated data each time new information arrives. For instance, Pina-Pardo et al. (2024a) present an approach that combines an insertion heuristic with a MILP model to adjust drone resupply operations online.

  - Optimization with integrated future information: Deterministic models can also integrate future information directly at each decision epoch. This can be done by solving a deterministic MILP using expected release dates at each epoch (see Li et al. (2024b)) or by constructing multiple scenarios and solving the resulting deterministic routing problem for each one; a consensus function is then used to select the final routing plan over all scenarios (Voccia et al., 2019; Côté et al., 2023; Lan et al., 2024; Li et al., 2024b).

  - Useful information for stochastic case: Deterministic solutions provide useful information that can guide the solution of the stochastic case. For example, Crainic et al. (2018) use reduced costs as-

sociated with non-basic variables in deterministic solutions to guide variable selection in stochastic formulations.

2.2.2 Classification scheme describing characteristics of SDD problems

In Figure 2.4, we provide a classification of SDD problem characteristics, namely, problem types, typical objective functions, decisions, and constraints.

For problem types, we follow the classification defined by Psaraftis et al. (2016), which is based on two dimensions: the evolution and quality of information (Pillac et al., 2013). The evolution of information relates to whether the input available to a planner changes over time, while the quality of information refers to the presence of uncertainty in inputs. Accordingly, we have the following problem types:

- Static and Deterministic: all inputs are known with certainty before determining solutions; there are no stochastic inputs.
- Static and Stochastic: solutions are determined a priori, but some input follows probabilistic patterns.
- Dynamic Without Future Knowledge: inputs are received dynamically, but no stochastic information is known about future inputs.
- Dynamic and Stochastic: inputs are updated dynamically with stochastic information about future events.

We enclose the four problem types in a yellow rectangle in Figure 2.4 to indicate that incorporating any source of uncertainty listed in Figure 2.3 affects how the problem is classified.

We focus on systems where SDD companies are the primary decision-makers. These companies typically need to decide whether to accept or reject incoming orders, which delivery options to offer in terms of locations and

transportation modes, how to price time slots, assign services to drivers, schedule departure times, and design routes (see the process flow connecting all these decisions in Section 2.2.1.1). Specifically, an order is considered *accepted* when the customer is offered an SDD service. While acceptance or rejection decisions are mostly made immediately, some models allow for postponing this decision, only rejecting unserved orders at the end of the day for next-day delivery via their own fleets or third-party companies (Klapp et al., 2018a). In Figure 2.4, we enclose these decisions in a blue rectangle to highlight that they can be made either sequentially or in an integrated way. This distinction has been previously discussed in Section 2.2.1.1 in relation to Figure 2.3. Additionally, in dynamic problems, decision epochs refer to moments when decision-making is triggered. These moments are typically set at fixed times during the day, whenever a new order arrives, whenever a vehicle returns to the depot, or as a combination of these events.

Regarding constraints, unlike conventional delivery systems, where vehicle capacity is a big concern, delivery deadlines take precedence in SDD due to lower order volumes and tight delivery times (Stroh et al., 2022). To meet these deadlines, SDD companies must carefully manage fleet operations while balancing the tradeoff between using their own fleet and outsourcing deliveries to third-party delivery companies, which can be more costly but increases flexibility (Voccia et al., 2019). Optimization models that address these tradeoffs are subject to several constraints, such as the number of available vehicles (single or multiple vehicles), predefined time windows, and limits on waiting, response, or service times. There are also context-specific constraints, such as pickup station capacity when using designated pickup stations, battery charging requirements for electric vehicles or drones, and order pickup range, which accounts for customer preferences regarding the maximum distance they are willing to travel to pick up their orders.

The papers reviewed mostly handle operational challenges, with optimization goals including optimizing customer service and financial aspects. Specifically, some objectives commonly explored, such as the minimization of delays and the maximization of the number of requests served, are proxies for maximizing customer service levels; other objectives, such as the maximization of profit/revenue and the minimization of costs, represent the companies' financial goals. Particularly, service level objectives vary based on problem settings. For example, Yildiz &

Savelsbergh (2019a) evaluate click-to-door time (the difference between the drop-off time and the order time) and ready-to-door time (the difference between the drop-off time and the food ready time) in meal delivery; Chen et al. (2023) define service level in terms of fairness, measured as the ratio of expected number of accepted requests per day to total requests.

At the tactical level, SDD companies must determine fleet size or capacities distributed across different service zones (see Section 2.7). As delineated in recent works (Stroh et al., 2022; Banerjee et al., 2022; Wu et al., 2023), primary tactical optimization questions include:

- Fleet sizing - What is the optimal fleet size and composition?
- Service region partitioning - How to partition a service region into vehicle routing zones?
- Service network design - How to determine the shipment paths in both temporal and spatial dimensions?

Typically, these models are constructed to capture the average system behavior considering sampled or forecasted demand (Stroh et al., 2022; Banerjee et al., 2022). While specific objectives vary by study, the overall goal falls into developing simplified models that facilitate efficient day-to-day operational activities.

2.3 Release dates and deadlines

As explained earlier, this section discusses routing problems with release dates and/or deadlines, which serve as the basis for many other SDD problems. We begin by introducing release dates and outlining the related questions. In Section 2.3.1, we present a mathematical model for the basic setting with deterministic input data. In Section 2.3.2, we categorize representative papers based on their use of MDPs to model these problems. Within each category, we present the papers in chronological order, grouping those with mutual influences. Given that our survey

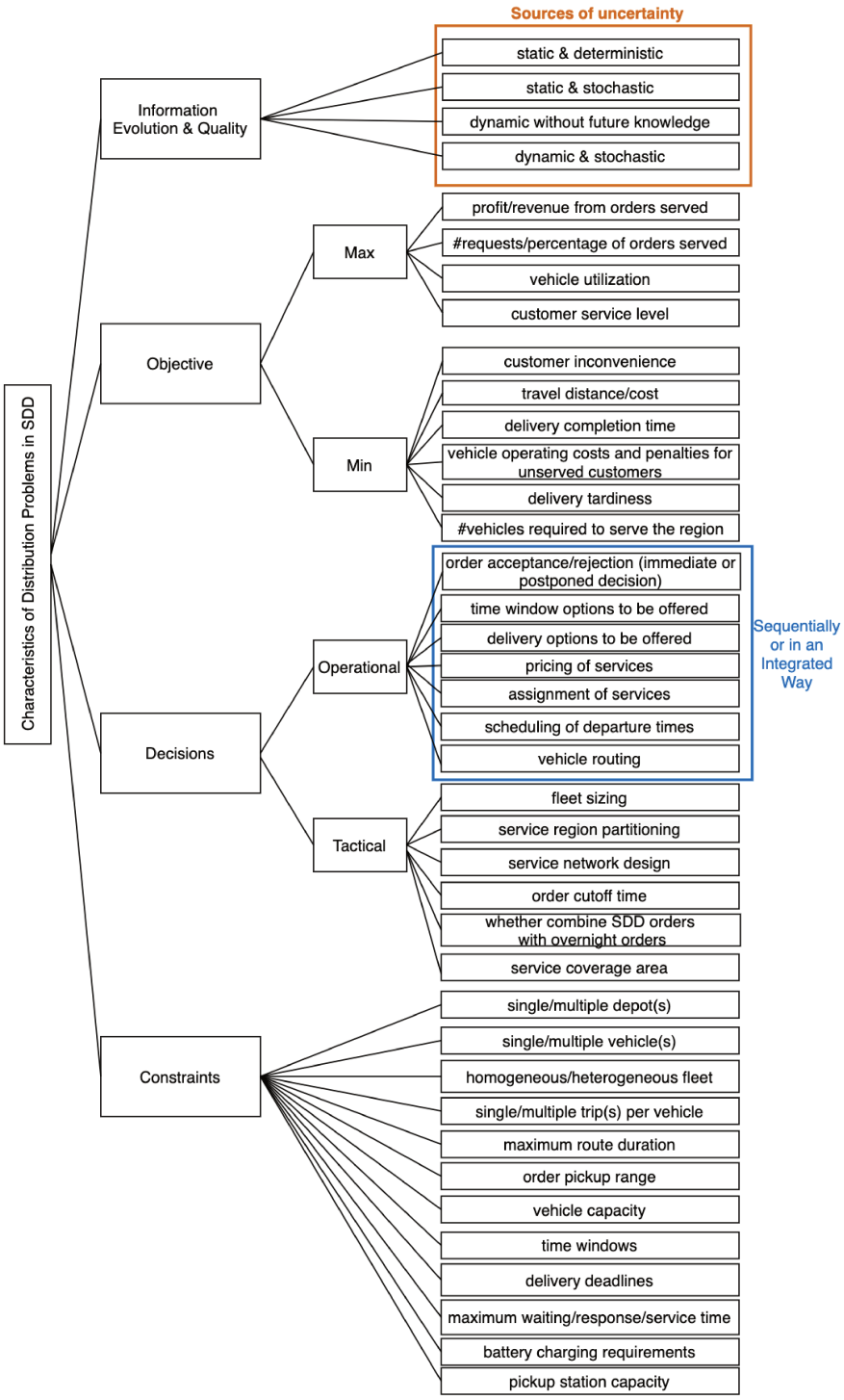


Figure 2.4: Classification scheme describing characteristics of SDD problems

focuses on papers addressing stochastic and dynamic problems, we briefly review studies centered on deterministic settings. For the papers investigating stochastic and dynamic problems, we highlight three key elements: the specific problem addressed, the solution approach used, and the results or findings obtained. We conclude this section by summarizing common findings and identifying gaps in the literature in Section 2.3.3.

In traditional delivery settings, all orders are available at the depot before deliveries. On the contrary, in SDD, orders arrive throughout the day. *Release dates* refer to the time when parcels become available at the depot for dispatch. Disclosure dates are when the decision-maker becomes aware of orders, also referred to as order times, as internet or system processing delays are typically neglected. The release date occurs after the disclosure date and corresponds to the time when the parcel is ready for dispatch (see Klapp et al. (2020)). If a parcel is already at the depot, which also functions as a warehouse, the release date coincides with the disclosure date. The products ordered online must be processed immediately by the distribution planners because of tight delivery deadlines. Hence, the fulfillment operations often happen simultaneously with the distribution operations. Moreover, the tight delivery time complicates logistics management as consolidation opportunities are significantly reduced. This raises challenging questions: Should the distribution planner wait before dispatching the vehicles for delivery to have more parcels available at the depot (and so to increase consolidation opportunities), or should the vehicles deliver the already available parcels in order to avoid missing the deadlines? Also, if the distribution planner decides to wait, how long?

To answer the above questions, a family of *routing problems with release dates and/or deadlines* has emerged in the recent literature. These problems aim to model the fundamental tradeoff between time and cost in SDD. Note that these problems are introductory to the problems presented in the following sections. In this section, we present the works that do not have overlapping features with the classes of problems analyzed in the following sections.

2.3.1 Mathematical formulation

This section focuses on the deterministic setting and presents the formulation for a basic routing problem with release dates and/or deadlines. As mentioned in Section 2.1, our goal is to analyze the emerging optimization problems in SDD and provide mathematical formulations for the fundamental problem in each class. This formulation might be considered as a reference for researchers addressing more sophisticated problems in the class and can be used as a baseline for comparisons or potential generalizations. For each class of problems analyzed, we present the underlying deterministic counterpart for the single-vehicle case.

In routing problems with release dates, we specifically focus on the deterministic Orienteering Problem with release dates (OP-rd) introduced by Li et al. (2024b). This is a basic problem whose setting and formulation can be easily adapted to different variants and more general cases. The problem is described as follows.

Let us assume that we have a set of parcels N with known release dates, $r_i \geq 0, i \in N$. Each parcel corresponds to a customer order and has to be delivered to the customer's place. The release date r_i represents either the time the order is placed (in case the parcel is already at the depot) or the time at which the parcel becomes available at the depot (for an earlier placed order). A single vehicle is available to perform delivery operations. The vehicle makes multiple routes that all start and end at the depot. During each route, the vehicle delivers a subset of the parcels available at the depot (thus, those associated with a release date not larger than the starting time of the route). We denote as V the set of locations including the customers' locations and the depot, denoted as 0. Set A is the set of arcs composed of any pair of nodes i, j in V. The travel distance between two nodes is denoted as $d_{ij} \geq 0$, where $i, j \in V$. Distances satisfy the triangle inequality. The goal of the problem is to maximize the number of customers that are served within the time horizon $[0, T_E]$, where T_E represents a *deadline* and can be interpreted as the end of the driver's shift. As mentioned above, the vehicle performs a set of routes $\mathcal{K}$ to deliver the orders within the deadline, where a route starts and ends at the depot and visits some customers in between. The solution is composed of $|\mathcal{K}|$ routes, where a trivial upper bound on the value of $|\mathcal{K}|$ is given by the number of

orders. A tighter upper bound is proposed by Li et al. (2024b). The routes $k \in \mathcal{K}$ are numbered starting from 0. A mathematical formulation for the problem is proposed by Li et al. (2024b). The problem is modeled as a MILP using the following decision variables: (i) y_i^k: binary variable equal to 1 if parcel i is delivered in route k, 0 otherwise; (ii) x_{ij}^k: binary variable equal to 1 if route k traverses arc (i,j), 0 otherwise; (iii) s_k: continuous variable representing the starting time of route k. The formulation for the OP-rd is:

$$\max \sum_{i \in N} \sum_{k \in \mathcal{K}} y_i^k \tag{2.1a}$$

$$\sum_{(i,j) \in A} x_{ij}^k = \sum_{(j,i) \in A} x_{ji}^k = y_i^k \qquad i \in V, k \in \mathcal{K} \tag{2.1b}$$

$$\sum_{i,j \in S} x_{ij}^k \leq \sum_{i \in S \setminus \ell} y_i^k \qquad S \subseteq N, |S| \geq 2, \ell \in S, k \in \mathcal{K} \tag{2.1c}$$

$$\sum_{k \in \mathcal{K}} y_i^k \leq 1 \qquad i \in N \tag{2.1d}$$

$$s_k \geq r_i y_i^k \qquad i \in N, k \in \mathcal{K} \tag{2.1e}$$

$$s_{k+1} = s_k + \sum_{(i,j) \in A} d_{ij} x_{ij}^k \qquad k \in \mathcal{K} \tag{2.1f}$$

$$s_{|\mathcal{K}|} = T_E \tag{2.1g}$$

$$x_{ij}^k, y_i^k \in \{0,1\} \qquad i,j \in V, k \in \mathcal{K}. \tag{2.1h}$$

The objective function (2.1a) maximizes the number of customers served. Equations (2.1b) are in- and out-degree constraints, inequalities (2.1c) are the generalized subtour elimination constraints, and (2.1d) state that each customer can be served at most once. Constraints (2.1e)–(2.1g) determine when each route starts. Specifically, constraints (2.1e) force a route to start after the arrival of parcels delivered on the route, constraints (2.1f) ensure that a new route starts when the previous route finishes, and constraints (2.1g) state that the last route ends at T_E.

The equality in constraints (2.1f) and (2.1g) is due to the fact that we can shift the waiting time to the start of the distribution without affecting the structure and the solution value so that any waiting time between consecutive routes and the end of the shift is avoided (see Archetti et al. (2018) for more details).

For an alternative formulation that shows good performance, consider the model proposed in Neira et al. (2020). This formulation utilizes auxiliary arcs instead of an upper bound $|\mathcal{K}|$. We decided to present Formulation (2.1), as it is a more natural and straightforward approach and easier to generalize than the model proposed in Neira et al. (2020).

Formulation (2.1) can be adapted to more general or similar problems. For example, instead of maximizing the number of parcels served, the objective could be to minimize the completion time for serving all customers (see Archetti et al. (2020a)) by forcing each customer to be visited (changing inequality constraints (2.1d) to equality) and removing constraints (2.1g). We can also handle the problem with multiple capacitated vehicles by adding capacity constraints and applying the constraints for each vehicle. Problems with additional side constraints, like delivery time windows, can also be easily integrated.

Most of the contributions mentioned in the following subsection deal with the stochastic and dynamic settings, in which the uncertainty is mostly related to release dates. The problem is then typically modeled as an MDP (Ulmer & Streng, 2019; Van Heeswijk et al., 2019; Archetti et al., 2020a; Ulmer, 2020; Li et al., 2024b). When dealing with dynamic and stochastic problems, Formulation (2.1) can be used in multiple ways: For competitive analysis (i.e., evaluating the solution with perfect information) or for finding a solution in each decision epoch using an approximation of future information. Both ways are applied in Li et al. (2024b), so we refer the reader to this paper for a reference about using a basic deterministic model to solve a stochastic and dynamic problem.

2.3.2 Literature review on SDD routing problems with release dates and deadlines

Some early contributions exist in studying routing problems where parcels arrive dynamically during the distribution process. Although the authors did not use the term "release date," the setting corresponds to the model introduced above. Using the concepts introduced in Section 2.2.2, we summarize the papers to give an overview and provide features of Model (2.1) as a point of reference in Table 2.1. The citations in Table 2.1 are listed chronologically. Most papers deal with dynamic and stochastic VRP or Traveling Salesman Problem (TSP), aiming to maximize the number of requests served or minimize travel distance/cost.

Table 2.1: Summary of papers on SDD routing problems with release dates and deadlines. The papers above the horizontal line do not use MDP-based models, while the papers below adopt MDP. The abbreviations: multiple depots (MD), single vehicle (SV), multiple routes per vehicle (MT), vehicle capacity (VC), time windows (TW), maximum waiting/response/service time (WT), delivery deadlines (DD, which denotes either the end of drivers' shift or the latest allowable time for delivering to customers), vehicle routing (route), scheduling of departure times (schedule).

Literature	Problem	Objective	Uncertainty	Inf. Evolution&Quality	Decision	Constraint
Azi et al. (2012)	VRP	max(profit collected from the orders served)	requests(arrival time, location)	dynamic & stochastic	accept/reject; route	TW; MT; DD
Ferrucci et al. (2013)	VRP	min(customer inconvenience)	requests(arrival time, location)	dynamic & stochastic	route	TW; DD; WT
Archetti et al. (2015)	TSP & VRP	min(travel distance)&min(completion time for serving all customers)	no	dynamic without future knowledge	route	DD; MT; SV
Ferrucci & Bock (2016)	VRP	min(customer inconvenience)	requests(arrival time, location)	dynamic & stochastic	route	TW; DD; WT
Cattaruzza et al. (2016)	VRP	min(travel distance)	no	static & deterministic	route; assignment of services	TW; VC; DD; MT
Reyes et al. (2018b)	TSP & VRP	min(travel distance)&min(completion time for serving all customers)	no	dynamic without future knowledge	route	TW; DD; MT; SV
Yao et al. (2019)	TSP	min(travel cost)	requests(location)	dynamic & stochastic	route	DD; MT;SV
Leung et al. (2023)	VRP	max(usage efficiency of vehicles) and min(the waiting time of orders)	requests(arrival time, location, weight)	dynamic & stochastic	schedule; route	DD; VC; WT
Ulmer (2017)	VRP	min(sum of delay over all customers)	requests(arrival time, location)	dynamic & stochastic	route	TW; DD; MD
Klapp et al. (2018a)	TSP	min(vehicle operating costs and penalties for the unserved)	requests(arrival time)	dynamic & stochastic	accept/reject; schedule; route	DD; MT;SV
Van Heeswijk et al. (2019)	delivery dispatching	min(travel distance)	requests(arrival time, demand quantity, order volume, location, dispatch window)	dynamic & stochastic	schedule; route	TW; VC; DD; MT
Voccia et al. (2019)	VRP	max(#requests served)	requests(arrival time)	dynamic & stochastic	accept/reject; schedule; route	TW; DD; MT
Ulmer et al. (2019b)	VRP	max(#requests served)	requests(arrival time, location)	dynamic & stochastic	accept/reject; preemptive return; route	DD; SV; MT
Klapp et al. (2020)	TSP	min(vehicle operating costs and penalties for the unserved)	requests(arrival time)	dynamic & stochastic	accept/reject; schedule; route	DD; SV; MT
Archetti et al. (2020a)	TSP	min(completion time for serving all customers)	requests(arrival time)	dynamic & stochastic	schedule; route	MT; SV
Chen et al. (2023)	VRP	max(overall service level and minimum regional service level)	requests(arrival time, location)	dynamic & stochastic	accept/reject; route	DD; MT
Côté et al. (2023)	VRP	min(#requests rejected) and min(routing cost)	requests(arrival time)	dynamic & stochastic	accept/reject; schedule; route; preemptive return	TW; DD; MT
Zhang et al. (2023)	VRP	max(#requests served)	requests(arrival time, demand quantity, location, service time)	dynamic & stochastic	accept/reject; schedule; route	DD
Lan et al. (2024)	VRP	min(travel distance)	requests(demand quantity, location, time window, service time)	dynamic & stochastic	schedule; route	TW; DD; VC
Li et al. (2024b)	OP	max(#requests served)	requests(arrival time)	dynamic & stochastic	accept/reject; schedule; route	DD; MT;SV
Model (2.1)	OP	max(#requests served)	no	static & deterministic	accept/reject; schedule; route	DD; SV

2.3.2.1 Non-MDP Models

This section presents studies that address routing problems with release dates and deadlines not modeled as MDP. These papers either do not model the transition dynamics for handling dynamic and stochastic environments or focus on deterministic problems.

Azi et al. (2012) study a problem arising in the distribution of online grocery orders. Orders arrive dynamically during the day, and the provider has to decide each time an order is placed whether to accept it or not. Each order is associated with a profit, and a fleet of vehicles is available to perform delivery operations. Each vehicle can perform multiple routes of a limited duration. The goal is to maximize the profit collected from the orders served. The authors propose a heuristic embedding an acceptance rule that takes into account potential future requests. The algorithm is compared against a myopic rule, and the results show that it provides a competitive advantage. Unlike the problem in Azi et al. (2012), where vehicles need to come back to the depot to take new parcels to deliver, Ferrucci et al. (2013) consider a daily distribution process evolving in real-time, specifically for applications where vehicles are loaded with sufficient quantities at the depot, avoiding the need to return for new requests, such as the delivery of newspapers. Customer requests arrive dynamically during the day. Each request is associated with a soft time window; a high penalty is paid if the service begins after the window. The problem is to decide at each specified decision time the route that assigns the set of unserved requests to vehicles to minimize the total customer inconvenience captured by a function of request response times. The authors exploit past request information to anticipate future requests. Ferrucci et al. (2013) assume that request arrivals on different days follow the same spatial and temporal structure. Thus, they use a single profile of past requests to anticipate future request arrivals. However, Ferrucci & Bock (2016) observe that different days follow different distributions, so they extend the single profile approach by applying multiple profiles for the same problem setting studied in Ferrucci et al. (2013). They identify k profiles by applying a k-means algorithm (MacQueen, 1967). During the day, request arrivals are analyzed in real time to map the request arrival structure to a profile. The computational results compared to the

previous single profile approach prove the superiority of the new approach.

The papers above consider multiple vehicles. On the other hand, Yao et al. (2019) consider a setting in which a single uncapacitated vehicle is dispatched from a local distribution center, with only a subset of parcel locations being known in advance. Additional parcels' locations become known while the vehicle performs deliveries, and preemptive returns are allowed. Preemptive return means that the vehicle might return to the depot before having delivered all parcels on board if convenient. The problem is modeled as a precedence-constrained asymmetric TSP with demand uncertainties. The uncertainty is addressed with robust optimization, and a MILP formulation is proposed.

Instead of focusing on solution approaches like the previous papers, Archetti et al. (2015) study the complexity of single and multiple VRP with release dates on two variants: minimizing the total distance traveled while respecting a deadline and minimizing the total time needed to complete the distribution. Although the problems are generally NP-hard, the authors show they are polynomially solvable when the underlying graph is a star or a line. Afterward, Reyes et al. (2018b) extend the problem settings of Archetti et al. (2015) by considering service guarantees, which sets customer-specific delivery deadlines. They propose alternative dynamic programs for VRP with release dates and show that the variants where customers are located on a half-line can be solved optimally in polynomial time.

To the best of our knowledge, Cattaruzza et al. (2016) is the first paper to use the term "routing problems with release dates." The authors study a two-level system where trucks deliver parcels to a local distribution center, with release dates known in advance. Vans then deliver parcels to customers within predefined time windows. The objective is to minimize routing costs while ensuring timely deliveries.

In the models presented so far, consolidation opportunities are implicitly exploited while considering certain objectives (like maximizing profit from the orders served, or minimizing delivery cost). Leung et al. (2023) propose a more explicit objective, addressing a delivery problem with multiple vehicles and pickup stations, which is

maximizing vehicle usage and minimizing order waiting times. Their approach clusters orders into groups, each served by one vehicle, optimizing two criteria: (i) temporal postponement to determine the best delivery shift for each request and (ii) spatial adjustment to modify the area served by each vehicle.

2.3.2.2 MDP-Based Models

Recent studies often model stochastic and dynamic SDD problems using MDPs, as they provide a structured framework for sequential decision making under uncertainty. This section presents works that use MDPs to model SDD problems with release dates and deadlines. The related works have appeared in the last ten years. As many related works have subtle but important differences in problem settings, we further divide them into two groups - single and multiple vehicle studies. We now review the most prominent contributions among these studies.

Single vehicle: In SDD, not all orders may be served by the end of the day. To deal with unserved orders, Klapp et al. (2018a) impose penalties and minimize the (expected) vehicle travel cost and penalties for unserved requests. In their model, dispatching decisions are made at fixed intervals, and no preemptive returns are allowed. Specifically, all requests have the same deadline, i.e., the end of the planning horizon. Additionally, it is assumed that all customers lie on a line segment and that the depot is located at one end of this segment. The authors first propose an integer programming model and a local search heuristic to solve the deterministic variant of the problem where all the release dates are known. Then, they use this approach to derive *a priori* solutions for the stochastic variant with predetermined routes. They further develop two approaches to obtain dynamic policies: a rollout policy and a fast heuristic modification to the initial solution. Unlike Klapp et al. (2018a), Ulmer et al. (2019b) focus on evaluating the value of allowing preemptive depot returns in a more general network structure. The goal is to maximize the expected number of served customers within the given planning horizon. They use VFA as a solution method. Through an extensive computational study, they show that preemptive returns are most effective when the returns occur late enough in the planning horizon, while they are much less effective in other cases. Archetti et al. (2020a) also study a single-vehicle problem with stochastic and dynamic release dates, with no deliv-

ery deadline or preemptive returns. The objective is to minimize the total time spent serving all customers. They propose a reoptimization approach that heuristically reduces the number of decision epochs. The optimization problem at each decision epoch is solved in two ways: through a deterministic model that takes as input the point estimation of release dates corresponding to the expected value and through a stochastic optimization model that considers the entire stochastic information on the release dates. At each reoptimization epoch, they determine the solution of the two models through an iterated local search. The computational results show that the stochastic model performs better than the deterministic one but is more time-consuming. The deterministic counterpart of their problem has been studied earlier by Archetti et al. (2018).

Besides, Li et al. (2024b) introduce the orienteering problem with stochastic and dynamic release dates, whose deterministic counterpart corresponds to the problem presented in Section 2.3.1. In their setting, the release dates are stochastic and follow a known distribution. As in Archetti et al. (2020a), no preemptive returns are allowed, and a delivery deadline is considered. The objective is to maximize the expected number of requests served before the deadline. The authors propose an exact branch-and-cut method to solve the optimization problem at each decision epoch (as opposed to the heuristic methods used by Archetti et al. (2020a)). Specifically, they propose an approximation of future states and derive decision policies through two VFA approaches. The first uses a consensus function (in the first VFA), while the second solves a two-stage stochastic MILP model, simultaneously taking multiple sampled scenarios into account (in the second VFA). The authors empirically demonstrate the advantage of embedding exact methods inside a reinforcement learning framework to improve the sequential decision-making process.

The papers (Klapp et al., 2018a; Ulmer et al., 2019b; Li et al., 2024b) allow for postponing acceptance/rejection decision of SDD services, only rejecting unserved orders at the end of the day. On the contrary, Klapp et al. (2020) study a problem where customers place orders over time. The dispatcher immediately decides whether to accept the order (and thus, guarantee that the parcel will be delivered within the day) or to deny the service. A single vehicle is used to perform deliveries, no preemptive returns are allowed, and a penalty is imposed for each denied

request. The goal is to determine the distribution plan that minimizes the expected sum of vehicle travel costs and penalties for denials. Vehicle routes are updated dynamically and serve each accepted delivery request by the end of the day. The solution approach is based on determining an a priori plan and then updating it with a rollout procedure.

Multiple vehicles: Ulmer (2017) studies a dynamic multi-vehicle, multi-depot routing problem with stochastic release dates. The problem setting includes *delivery deadlines*. Contrary to most papers, where hard deadlines are imposed (i.e., the customer is considered "served" only if the delivery happens before the given deadline), Ulmer (2017) allows for *soft deadlines* (i.e., in case the order is not delivered before the deadline, the customer experiences a delay). The goal is to minimize the expected average delay per customer. The author performs a simulation study on the Iowa City area, concluding that narrow delivery deadlines cause higher delivery costs. Voccia et al. (2019) study a single-depot routing problem with a hard delivery deadline. The customer locations are known in advance, but the release dates are subject to uncertainty. It is assumed that the arrival rate and the distribution of release dates for each order are given. Moreover, each order is associated with a (deterministic) deadline or a time window during which the delivery must occur. The goal is to maximize the expected number of requests served within the planning horizon. By incorporating information about future requests into routing decisions, the authors derive an analytical formula that identifies when it is beneficial for vehicles to wait at the depot. They also present a heuristic solution approach based on scenario sampling combined with a consensus function. Côté et al. (2023) study a similar problem, where the primary objective is to minimize the number of rejected requests, while the secondary objective is to minimize the total routing cost. Besides, preemptive returns are allowed. To address the problem, the authors propose a branch-and-regret (B&R) heuristic, which incorporates scenario sampling techniques, consensus functions, and a branching scheme. Through testing on the same instances utilized by Voccia et al. (2019), the authors show that their B&R performs much faster, achieving a higher number of served requests.

Van Heeswijk et al. (2019) present another variant of the multi-vehicle delivery dispatching problem. The num-

ber of orders arriving, volume, destinations, and dispatch windows are uncertain. Every order must be shipped within its dispatch window. Dispatch operations are performed using a fleet of primary and secondary vehicles, whereas secondary ones are more expensive. However, it is assumed that all orders can be dispatched within the planning horizon. The goal is to minimize the expected dispatching costs. To handle large instances, the authors propose a VFA algorithm with the approximation of routing cost. Lan et al. (2024) study a similar problem but assume a sufficient number of primary vehicles to serve all requests on time, ensuring that vehicle dispatch decisions in the current epoch do not impact decisions in future epochs. They propose an iterative solution construction procedure based on a scenario sampling algorithm, classifying requests into three categories: those to be dispatched, postponed, or determined later. In each scenario, a static routing problem is solved. By testing on instances from the EURO Meets NeurIPS 2022 Vehicle Routing Competition, they show that the algorithm performance nearly matches that of the winning ML-based approach.

Meanwhile, concerns about the fairness of the delivery services are rising as faster delivery services are spreading out. In 2016, Amazon was accused of racial disparities in its Prime SDD zones in several U.S. cities (Howland, 2016). To address the unfairness in SDD service availability, Chen et al. (2023) first partition the service area and then define fairness as the minimum service rate over the resulting regions. They model the problem as a multi-objective MDP with a weighted sum of two objectives: maximizing the overall service level (utility) and maximizing the minimum service rate across all regions (fairness), controlled by a factor ranging from 0 to 1. No fairness is considered when the factor is valued at 0; when it is at 1, no utility is considered. They develop a deep Q-learning approach and show its effectiveness in moderating unfairness for customers from different geographic areas.

The papers discussed above focus on problems with release dates, where vehicles are dispatched from depots to collect parcels. On the other hand, Zhang et al. (2023) study applications without release dates but with deadlines where vehicles only need to travel directly to customers' locations. Customer requests arrive dynamically and stochastically. The decision-maker aims to maximize the number of served requests by deciding in real time whether to accept or reject each request. If accepted, the request is assigned to a vehicle, and the route is adjusted to

accommodate it. The problem is modeled as a multi-stage optimization model and solved using scenario sampling, with knapsack-based approximations to estimate the expected cost-to-go. Computational studies based on a real street network show the effectiveness of the method.

2.3.3 Common findings and gaps

For stochastic and dynamic SDD routing problems with release dates and deadlines, a key idea is to anticipate future requests to improve decision-making processes. Early studies exploited machine learning (ML) techniques to anticipate the knowledge of future requests, such as their number, arrival time, and locations, based on past information (see Ferrucci et al. (2013) and Ferrucci & Bock (2016)). More recent works have increasingly adopted scenario sampling methods to anticipate requests. As solution quality and running time depend significantly on the precision and the number of scenarios generated, future research could explore combining the two methods to generate high-quality solutions.

2.4 Time window options for attended home delivery

In this section, we focus on distribution problems on time window options. We begin by highlighting their importance and the challenges they bring to SDD. Next, we present a mathematical model for a deterministic environment in Section 2.4.1. We then discuss recent works that tackle the challenges of deciding time slots or pricing options in Section 2.4.2. Finally, we summarize this section by listing common findings and identifying gaps in the literature in Section 2.4.3.

Many SDD companies offer customers the flexibility to decide when, where, and how their parcels should be delivered. This generates complex decision-making challenges, such as which time window options to offer, how to price different time window options, where to deliver when there are multiple possible delivery locations, and

how to coordinate different transportation methods. In this section, we narrow our focus on time window options and pricing. Because of the importance of the temporal and pricing aspects in SDD, we provide a mathematical formulation for time window options, which warrants a separate and dedicated section away from the next one that discusses location and transportation options.

In a UK consumer home delivery report, 1000 respondents were questioned on what they thought about the length of the delivery time slot (nShift, 2022). Over 40% of the respondents stated they were willing to wait for a maximum of 2 hours, while less than 10% would wait for more than 4 hours. This underlines the pressure from online customers on service quality: not only do they require fast deliveries (as mentioned in the former section), but also narrow time windows for delivery.

To avoid or reduce delivery failures, many researchers recently investigated the option of offering multiple time windows to customers in attended home delivery (AHD) services. This time window option is often applied in online grocery ordering. For each request, the customer and the grocery retailer must mutually agree on a time window for delivery. The process imposes computational challenges as the proposal of time window slots should be as fast as possible (within milliseconds) to ensure a smooth booking process. Besides, the combination of SDD and narrow deadlines is economically challenging for SDD companies (Ram, 2015). Thus, another emerging research trend is studying *delivery time options with pricing*, i.e., customers are offered different delivery times associated with different prices (the tighter the delivery time, the higher the price).

2.4.1 Mathematical formulation

As done in the former section, we present the mathematical formulation of a basic problem when dealing with the delivery deadline options. This formulation might be used as a benchmark or a starting point to deal with more complicated problems in this class.

The formulation is built on the problem setting introduced by Ulmer (2020) but considering a single uncapacitated vehicle and a deterministic environment, i.e., the two unknown factors in Ulmer (2020) - customer release dates and customer choice behavior based on willingness-to-pay (WTP) - are assumed to be known here. We describe the problem as follows.

A set of customers N orders goods during the day, and we denote the order time of each customer i as r_i (which might be seen as the release date introduced in the former section). V is the set of locations, including the customers' locations N and depot 0, i.e., $V = N \cup \{0\}$. A set of same-day delivery deadline options Δ is available. The corresponding deadline of each option is denoted with D_δ ($\delta \in \Delta$). For each option $\delta \in \Delta$, the SDD company knows the highest amount a customer i is willing to pay (also known as the WTP), denoted with U_δ^i and has to determine a customer-specific price p_δ^i. If p_δ^i is smaller than U_δ^i, the option δ is feasible for customer i. If there are no feasible options, the customer will be moved to the next-day delivery. Note that, as we are in a deterministic setting, for each option δ that is feasible for customer i, i.e., $p_\delta^i \leq U_\delta^i$, p_δ^i should always be equal to U_δ^i in order to maximize the revenue. In this way, we can omit the decision variable p_δ^i as the offered price will be set according to the delivery option, which maximizes the provider's revenue. Thus, the provider selects the delivery option to be offered to each customer and builds the delivery routes for the vehicle so that the revenue from the selected delivery options is maximized. Customers served on the following day are not considered in the delivery plan of the day. The vehicle makes at most $|\mathcal{K}|$ routes to serve the customers. All routes start from depot 0, serve a set of customers, and go back to the depot. As in Formulation (2.1), a solution consists of up to $|\mathcal{K}|$ routes performed by the vehicle. The goal is to maximize the revenue from the selected delivery options within $T = [0, T_E]$.

The following decision variables are used: (i) $y_{i\delta}^k$: binary variable equal to 1 if customer i is served in route k with deadline option δ, 0 otherwise; (ii) x_{ij}^k: binary variable equal to 1 in case arc (i, j) is traversed in route k, $i, j \in V$; (iii) a_i: continuous variable representing the arrival time at customer i; (iv) s_k: continuous variable representing the starting time of route k; (v) e_k: continuous variable representing the ending time of route k. Furthermore, we

introduce auxiliary variables $y^k_{0\delta}$ with $U^0_\delta = 0$, for each $\delta \in \Delta$. The problem formulation is then:

$$\max \sum_{k\in\mathcal{K}} \sum_{\delta\in\Delta} \sum_{i\in N} U^i_\delta y^k_{i\delta} \tag{2.2a}$$

$$\sum_{k\in\mathcal{K}} \sum_{\delta\in\Delta} y^k_{i\delta} \leq 1 \qquad i \in N \tag{2.2b}$$

$$\sum_{\delta\in\Delta} y^k_{0\delta} \leq 1 \qquad k \in \mathcal{K} \tag{2.2c}$$

$$\sum_{(i,j)\in A} x^k_{ij} = \sum_{(j,i)\in A} x^k_{ji} = \sum_{\delta\in\Delta} y^k_{i\delta} \qquad i \in V, k \in \mathcal{K} \tag{2.2d}$$

$$a_i \geq r_i \qquad i \in N \tag{2.2e}$$

$$a_i \leq r_i + \sum_{\delta\in\Delta} \sum_{k\in\mathcal{K}} D_\delta y^k_{i\delta} \qquad i \in N \tag{2.2f}$$

$$a_j \geq a_i + d_{ij} \sum_{k\in\mathcal{K}} x^k_{ij} - M(1 - \sum_{k\in\mathcal{K}} x^k_{ij}) \qquad i,j \in N, i \neq j \tag{2.2g}$$

$$a_j \geq s_k + d_{0j} x^k_{0j} - M(1 - x^k_{0j}) \qquad j \in N, k \in \mathcal{K} \tag{2.2h}$$

$$a_i \leq e_k - d_{i0} x^k_{i0} + M(1 - x^k_{i0}) \qquad i \in N, k \in \mathcal{K} \tag{2.2i}$$

$$e_k = s_k + \sum_{(i,j)\in A} d_{ij} x^k_{ij} \qquad k \in \mathcal{K} \tag{2.2j}$$

$$s_k \geq r_i \sum_{\delta\in\Delta} y^k_{i\delta} \qquad i \in N, k \in \mathcal{K} \tag{2.2k}$$

$$s_{k+1} \geq s_k + \sum_{(i,j)\in A} d_{ij} x^k_{ij} \qquad k \in \mathcal{K} \tag{2.2l}$$

$$s_{|\mathcal{K}|} \leq T_E \tag{2.2m}$$

$$x^k_{ij}, y^k_{i\delta} \in \{0,1\} \qquad i,j \in V, \delta \in \Delta, k \in \mathcal{K} \tag{2.2n}$$

$$s_k, a_i, e_k \geq 0 \qquad i \in V, k \in \mathcal{K} \tag{2.2o}$$

The objective function (2.2a) maximizes the revenue. Constraints (2.2b) guarantee each customer is delivered at most once with one of the SDD options. Constraints (2.2c) limit the depot to be visited at most once in each route. Constraints (2.2d) are degree constraints. Constraints (2.2e) and (2.2f) ensure that the delivery time of a parcel is after its order time but before the promised deadline. Constraints (2.2g) set delivery times of customers visited consecutively in a route. Constraints (2.2h) and (2.2i) fix the starting and ending time of each route with respect to the visiting time of the first and last customer on the route. Constraints (2.2j) calculate the ending time of each route on the basis of the starting time and the duration of the route. Constraints (2.2k) ensure a route can only start after the order time of the requests to be served. Constraints (2.2l) state that a new route starts after the end of the preceding route. Constraints (2.2m) ensure the ending time is within the time horizon. Note that constraints (2.2g) are sufficient to prevent subtours. However, generalized subtour elimination constraints (2.2p)

$$\sum_{i,j \in S} x_{ij}^k \leq \sum_{i \in S \setminus \ell} \sum_{\partial \in \Delta} y_{i\partial}^k, \qquad S \subseteq N, |S| \geq 2, \ell \in S, k \in \mathcal{K} \tag{2.2p}$$

can be introduced to strengthen the formulation.

This formulation can be adapted to handle the case of multiple capacitated vehicles by applying the constraints above for each vehicle and adding capacity constraints.

The problem considered above is deterministic. The settings in the contributions mentioned in the following subsection are mostly stochastic and dynamic, with uncertain release dates or customers' preferences. The problem is often modeled as an MDP (Ulmer, 2020; Klein & Steinhardt, 2023). When considering the uncertain arrival time of customer orders and customers' individual preferences (WTP), the Formulation (2.2) can be used to evaluate the solution with perfect information for competitive analysis, or it can be embedded into the stochastic and dynamic approach.

2.4.2 Literature review on delivery time slots with pricing

Table 2.2: Summary of papers on time window options for attended home delivery. The abbreviations: multiple routes per vehicle (MT), time windows (TW), single vehicle (SV), and delivery deadlines (DD), vehicle routing (route).

Literature	Problem	Objective	Uncertainty	Inf. Evolution&Quality	Decision	Constraint
Ulmer (2020)	VRP	max(revenue)	requests(arrival time), customer choice	dynamic & stochastic	time window options; pricing of services; route	DD; MT
Klein & Steinhardt (2023)	VRP	max(profits)	requests(arrival time), customer choice	dynamic & stochastic	time window options; pricing of services; route	TW; DD; MT
Banerjee et al. (2024)	tactical design of SDD system	max(customer satisfaction); max(profits)	requests(arrival time, location), customer choice	dynamic & stochastic	order cutoff time; pricing of services	DD; MT
Model (2.2)	OP	max(revenue)	no	static & deterministic	delivery options; route	DD; SV; MT

In many AHD applications, once a customer initiates a delivery request, the provider not only derives anticipatory decisions on which delivery time slots should be offered to the customer but may also decide to charge the customer for some of the offered time slots. Indeed, as Ulmer (2017) has shown, delivering orders within a short time is costly. The author quantifies the impact of delivery deadlines on delivery costs and finds that a deadline of 60 minutes may double the number of vehicles and working time needed to perform the service compared to a delivery service with a 3- or 4-hour deadline. However, not many users are willing to pay much for instant deliveries. Based on a survey of over 4700 respondents in China, Germany, and the United States (Joerss et al., 2016), about 25% of consumers (especially young people) are willing to pay premiums for the privilege of SDD or instant delivery. Nevertheless, only 2% said they would pay sufficiently more to make the instant delivery viable. Thus, careful pricing management is necessary to meet the delivery costs while guaranteeing a certain level of customer satisfaction. Following the concepts introduced in Section 2.2, we summarize the papers to give an overview and provide features of Model (2.2) as a point of reference in Table 2.2. We see that, besides request arrival time, customer choice is a prevalent uncertainty factor considered, as these problems involve direct customer interactions.

Ulmer (2020) studies how to price different deadline options for customers arriving dynamically. They show how dynamic pricing can considerably increase companies' revenues and the number of requests served within the time horizon. The focus is on the uncertainty of customer choice behavior, modeled through a distribution function of the customer's WTP for each delivery option. The authors first model the problem as an MDP and then propose an anticipatory pricing and routing policy, based on VFA, that incentivizes customers to select delivery deadline options that are most convenient for the routing plan. The pricing policy considers both the current revenue and the opportunity cost. The latter is defined as the difference in future revenue between the case in which the customer accepts an option and when she does not. The state-dependent opportunity cost per customer and option is calculated to consider the uncertain WTP. Then, an approximation through state space aggregation is implemented to deal with the curse of dimensionality.

Similarly to Ulmer (2020), Klein & Steinhardt (2023) optimize the delivery time slots and prices offered to each incoming customer request. The main difference between the two studies lies in the underlying solution approach. Unlike Ulmer (2020), whose methodology relies on a learning-based VFA, Klein & Steinhardt (2023) use scenario sampling to derive an online route and a choice-based pricing optimization problem with discrete price points to determine options pricing. Interestingly, they find that maximizing the number of requests served does not necessarily lead to optimal contribution margins due to the associated high delivery costs. Rather than dynamically pricing each new order, Banerjee et al. (2024) study pricing decisions at the tactical level by setting static prices for SDD. Similar to Ulmer (2020), they assume next-day delivery option is free. However, unlike Ulmer (2020) and Klein & Steinhardt (2023), where next-day delivery orders are either handled by a separate fleet from the SDD orders or not considered, they propose a system that uses the same delivery fleet to fulfill both SDD and next-day deliveries. They consider two objectives: first, they aim to maximize customer satisfaction by maximizing the number of next-day deliveries that arrive a day early, based on empirical evidence that exceeding customer delivery time expectations can grow market share; second, they aim to maximize total profit by offering discounts on SDD prices for early-order customers. They present a deterministic model that replaces stochastic order arrivals with

fluid order approximations. Their simulation suggests that their system is robust under stochastic conditions.

2.4.3 Common findings and gaps

The research studies on dynamic pricing of delivery time slots are rather limited. The first two papers have developed solution approaches for managing simultaneous decisions on customer-dependent pricing and vehicle routing in real time, such as the learning-based VFA in the first paper and the scenario sampling approach combined with a choice-based pricing optimization in the second. However, many interesting aspects remain to be studied in this area. For example, most papers on dynamic pricing only study short-term payoffs, i.e., the revenue obtained in a short time period. For long-term payoff, more studies are needed to determine the best pricing policy.

2.5 Delivery options

This section analyzes routing problems with delivery options arising in SDD services. We start by defining the concept of delivery options. Next, we focus on delivery location options in Section 2.5.1, which includes a MILP model with deterministic input data in Section 2.5.1.1 and a review of related literature in Section 2.5.1.2. Then, we discuss non-traditional transportation options in Section 2.5.2, including the usage of drones and crowdshipping. As in previous sections, we outline three key elements for each stochastic and dynamic paper: the specific problem addressed, the solution approach taken, and the results or findings obtained. We conclude the section with common findings and gaps in Section 2.5.3.

The term *delivery option* indicates both the case where multiple options are offered to customers as to where parcels should be delivered and the case where different options are available concerning how the parcels are delivered (different transportation means or different cohorts of drivers).

E-retailers and shipping companies often provide different options to customers concerning where parcels should be delivered (office, home, lockers, or car trunk (Rubin, 2018)). Some shared delivery locations are, for instance, pickup stations (parcel locker) and shops (Savelsbergh & Van Woensel, 2016). They work as secured and unattended customer locations, where couriers deliver packages for later pickup by customers. In this way, customers have more flexibility in picking up their parcels than waiting at home. Moreover, drivers can deliver many parcels to a pickup station instead of serving customers individually, thus reducing operating costs and pollution. Furthermore, failed deliveries are reduced, and this makes the constrained capacity in parcel-delivery networks less tight (Barbee et al., 2021).

As for transportation means, autonomous vehicles can be considered a novel technology to face the increasing shortage of drivers. In addition, crowdshipping is an emerging delivery strategy where ordinary people transport parcels instead of companies' drivers, against compensation. The main advantage of crowdshipping is that it allows the dynamic adjustment of the delivery capacity according to demand fluctuations without incurring fixed costs associated with hiring regular drivers.

Overall, delivery options provide more flexibility, but, at the same time, they increase the planning complexity, both when they refer to location options (multiple locations are offered) or transportation options (multiple means of transportation or different cohorts of drivers are available).

2.5.1 Where to deliver

This section focuses on problems where multiple delivery location options are available.

2.5.1.1 Mathematical formulation

We present a MILP model based on the setting of the problem studied in Ulmer & Streng (2019) with some modifications. Specifically, as in the former sections, we consider a single uncapacitated vehicle and we assume to have perfect information, while in Ulmer & Streng (2019) orders are unknown. Also, the time at which orders are placed is known beforehand. The problem is described as follows.

A set of customers N orders goods online over a discrete time horizon $T = (0, 1, \ldots, T_E)$. We denote the order time (release date) of each customer with r_n, $n \in N$. Deliveries are not made to customers' locations but to a set of pickup stations, where multiple parcels can be delivered in a single trip. Specifically, we denote as $\mathcal{P}$ the set of pickup stations and as 0 the depot. $\mathcal{P}_0$ is the union of $\mathcal{P}$ and 0. We assume that each customer can be served from any of the stations that are within a travel radius centered around the customer's location. We denote as P_n the set of stations that can be used to serve customer n. The delivery is performed by a single uncapacitated vehicle. Consistently with the assumption made in Ulmer & Streng (2019), we assume that the vehicle only travels from the depot to a pickup station and back. This type of trip is called direct trip. Thus, a trip corresponds to a visit to a pickup station, and no routing decision is involved. The capacity of pickup station $i \in \mathcal{P}$ during the entire time horizon is c_i, and the newly freed space due to orders being picked up is not considered. Each parcel occupies one unit of station capacity. The travel time from the depot to a pickup station i and from a pickup station i to a depot is given by d_{0i} and d_{i0} respectively, $i \in \mathcal{P}$. A solution consists of up to $|\mathcal{K}|$ trips performed by the vehicle, where the number of $|\mathcal{K}|$ is not greater than $|N|$. The objective is to minimize the sum of delivery times, i.e., the sum over all customers of the difference between the *earliest pickup time* (the time when the order is delivered to a pickup station) and the order time. In order to serve as many customers as possible before the deadline T_E, a big penalty value is added to the earliest pickup time in case a customer is not served.

The mathematical formulation of the problem makes use of the following decision variables: (i) y_{ni}^k: binary variable equal to 1 if the order of customer n is served in trip k to pickup station i, 0 otherwise; (ii) g_n: binary

variable equal to 1 if customer n is served, 0 otherwise; (iii) s_k: continuous variable representing starting time of trip k; (iv) t_n: continuous variable representing earliest pickup time of order n; (v) z_i^k: binary variable equal to 1 if pickup station i is visited in trip k, 0 otherwise. The formulation is the following:

$$\min \sum_{n \in N} (t_n - r_n) \tag{2.3a}$$

$$\sum_{i \in P_n} \sum_{k \in \mathcal{K}} y_{ni}^k = g_n \qquad n \in N \tag{2.3b}$$

$$\sum_{i \in \mathcal{P} \setminus P_n} y_{ni}^k = 0 \qquad n \in N, k \in \mathcal{K} \tag{2.3c}$$

$$\sum_{k \in \mathcal{K}} \sum_{n \in N} y_{ni}^k \leq c_i \qquad i \in \mathcal{P} \tag{2.3d}$$

$$t_n \geq M(1 - g_n) \qquad n \in N \tag{2.3e}$$

$$c_i z_i^k \geq \sum_{n \in N} y_{ni}^k \qquad i \in \mathcal{P}, k \in \mathcal{K} \tag{2.3f}$$

$$\sum_{n \in N} y_{ni}^k \geq z_i^k \qquad i \in \mathcal{P}, k \in \mathcal{K} \tag{2.3g}$$

$$s_k \geq \sum_{i \in P_n} r_n y_{ni}^k \qquad n \in N, k \in \mathcal{K} \tag{2.3h}$$

$$s_{|\mathcal{K}|} \leq T_E \tag{2.3i}$$

$$s_{k+1} \geq s_k + \sum_{i \in \mathcal{P}} (d_{0i} + d_{i0}) z_i^k \qquad k \in \mathcal{K} \tag{2.3j}$$

$$t_n \geq s_k + \sum_{i \in P_n} d_{0i} y_{ni}^k - (1 - \sum_{i \in P_n} y_{ni}^k) M \qquad n \in N, k \in \mathcal{K} \tag{2.3k}$$

$$\sum_{i \in \mathcal{P}} z_i^k \leq 1 \qquad k \in \mathcal{K} \tag{2.3l}$$

$$y_{ni}^k, z_i^k, g_n \in \{0, 1\} \qquad n \in N, i \in \mathcal{P}_0, k \in \mathcal{K} \tag{2.3m}$$

$$s_k, t_n \geq 0 \qquad n \in N, k \in \mathcal{K}. \tag{2.3n}$$

The objective function (2.3a) minimizes the service time over all customers. Constraints (2.3b) limit each customer to be served at most once, and (2.3c) ensure that customers are not assigned to pickup stations outside the radius. Inequalities (2.3d) establish the maximum capacity for each pickup station while (2.3e) associate a large penalty value with customers who are not served. Constraints (2.3f) state that a customer can be served by a station in a trip only if there is a delivery to the station and the capacity is not exceeded. Inequalities (2.3g) guarantee that a pickup

station is not visited in a trip when no parcel is delivered to the station and (2.3h) guarantee that each trip can only serve customers after order time. Constraints (2.3i) state that the last trip must finish before the deadline and (2.3j) set the starting time of each trip with respect to the previous trip. Constraints (2.3k) set the earliest pickup time for each customer where M is a big value. Finally, (2.3l) impose that a single pickup station can be visited in each trip.

The difficulty of the problem lies in satisfying simultaneously the capacity of pickup stations and the deadline. In a more general case, instead of considering direct trips only, the formulation can be adapted to include routing among multiple pickup stations in the same trip. The new problem shares similarities with the Generalized Traveling Salesman Problem (see Laporte et al. (1996)). The formulation is presented below.

First, we introduce an additional class of decision variables a_i^k representing the arrival time at the pickup station i in trip k, $i \in \mathcal{P}$. In addition, x_{ij}^k are binary variables equal to 1 if arc (i,j) is traversed in trip k, with $i,j \in \mathcal{P}_0$. We denote as A the set of arcs formed by any two nodes i,j in $\mathcal{P}_0$. Note that, in our model, variable z_i^k is defined not only for pickup stations but also for the depot, i.e., $i \in \mathcal{P}_0$. The new formulation reads as follows.

$$\min \sum_{n \in N} (t_n - r_n) \tag{2.4a}$$

$$(g_n, y_{ni}^k, z_i^k, s_k, t_n) \text{ satisfies (2.3b)-(2.3i)}$$

$$\sum_{(i,j)\in A} x_{ij}^k = \sum_{(j,i)\in A} x_{ji}^k = z_i^k \qquad i \in \mathcal{P}_0, k \in \mathcal{K} \tag{2.4b}$$

$$s_{k+1} \geq s_k + \sum_{(i,j)\in A} d_{ij} x_{ij}^k \qquad k \in \mathcal{K} \tag{2.4c}$$

$$s_k = a_0^k \qquad k \in \mathcal{K} \tag{2.4d}$$

$$a_j^k \geq a_i^k + d_{ij} x_{ij}^k - (1 - x_{ij}^k)M \qquad i \in \mathcal{P}_0, j \in \mathcal{P}, k \in \mathcal{K} \tag{2.4e}$$

$$t_n \geq a_i^k - (1 - y_{ni}^k)M \qquad n \in N, i \in P_n, k \in \mathcal{K} \tag{2.4f}$$

$$x_{ij}^k, y_{ni}^k, z_i^k, g_n \in \{0,1\} \qquad i,j \in \mathcal{P}_0, k \in \mathcal{K} \tag{2.4g}$$

$$s_k, t_n, a_i^k \geq 0 \qquad i,j \in \mathcal{P}_0, k \in \mathcal{K} \tag{2.4h}$$

Constraints (2.4b) are degree constraints and state that a parcel can be delivered to a station only when the station is visited. Constraints (2.4c) fix the starting time of consecutive trips while (2.4d) set the starting time of trip k as the time at which the vehicle leaves from the depot. Inequalities (2.4e) link the arrival times of nodes visited consecutively in a trip and (2.4f) link pickup times with arrival times. Similarly as in Formulation (2.2), subtours are eliminated through (2.4e) and subtour elimination constraints (2.4i)

$$\sum_{i\in S, j\in \mathcal{P}_0\setminus S} x_{ij}^k \geq z_l^k, \qquad S \subseteq \mathcal{P}_0, 0 \in S, l \in \mathcal{P} \setminus S, k \in \mathcal{K} \tag{2.4i}$$

can be introduced to strengthen the formulation.

As for the former cases, Formulations (2.3) and (2.4) can be adapted to solve similar problems. For instance, instead of minimizing the service time, the objective function can be maximizing the number of requests served before the deadline (which is implicitly considered in the current formulations by fixing the value of M). Also, one might want to consider cases without deadlines and where all customers must be served by searching for the minimum-cost solution.

To deal with dynamic and stochastic problems, i.e., customers' requests arriving dynamically, Formulations (2.3) and (2.4) can be handled similarly as mentioned in Section 2.3.1. Not only can it be used as a benchmark to assess online dispatching algorithms (Yildiz & Savelsbergh, 2019a; Pina-Pardo et al., 2024a) but also it can be embedded in the solution of the stochastic and dynamic problem and combined with a fast heuristic to adjust plans dynamically (Pina-Pardo et al., 2024a).

2.5.1.2 Literature review on delivery location options in SDD

Providing various delivery options is a way to reduce delivery failures. The related research problems span from determining the locations and layouts of pickup stations (Kahr, 2022) to selecting a delivery place for each request (Zhou et al., 2018; Dumez et al., 2021; Tilk et al., 2021; Mancini & Gansterer, 2021; Dragomir et al., 2022). There are different contributions to problems involving multiple delivery locations. However, studies that consider dynamic customer arrivals in SDD services are still limited.

Ulmer & Streng (2019) is the first study that combines pickup stations and autonomous vehicles for SDD. Each order is associated with a preferred pickup station and adjacent stations (called neighborhood) within a certain travel time around the preferred station are also accepted. The problem is to decide where and when to dispatch a vehicle and which parcels to deliver on each dispatch. Autonomous vehicles perform deliveries from the depot to stations, with only direct trips considered. The study addresses the tradeoff between consolidation and fast dispatches by defining a threshold specifying the minimum number of parcels that can be shipped in each dispatch. To determine the threshold, the authors propose a set of PFA approaches. In the computational section, the authors analyze three types of neighborhoods: none, normal, and large. They find that increasing neighborhood size decreases the percentage of deliveries to the preferred stations, but the service level improves.

Deliveries to pickup stations increase delivery efficiency via consolidation. A recent paper by Galiullina et al. (2024) considers the possibility of using a monetary incentive to steer customers from home delivery to pickup

station delivery. Different from the usual order-based fulfillment systems, where the assembly and delivery decision is made at the time when an order arrives (such as in Ulmer & Streng (2019)), in Galiullina et al. (2024), customers first place their orders and indicate the delivery locations; then, after a cutoff time, the decision-maker selects some customers to incentivize them to have their parcels delivered to the nearest pickup point. Customers may reject the offer and stick with home delivery. The delivery routes are made after all customers indicate their final delivery locations. The authors model the problem as a two-stage stochastic program and develop an exact branch-and-bound algorithm and several heuristics. They show the benefits of using such a demand steering approach against other policies.

2.5.2 Literature review on non-traditional transportation modes

Beyond the choice of delivery locations, SDD services often provide various options for transportation modes, including traditional vehicles with dedicated drivers, drones, and crowdshipping. Each mode has its cost structure, operational constraints, and environmental impact, making it crucial to select the appropriate option based on specific delivery scenarios. In the following, we focus on non-traditional modes, particularly on the use of drones and crowdshipping in SDD. We first divide the papers into those focused on drones and those on crowdshipping and then present them chronologically within each category, grouping those with mutual influences. Also, as Section 2.2.2, we summarize the papers presented in Sections 2.5.1.2 and 2.5.2 to give an overview of different features in Table 2.3, which includes Formulations (2.3) and (2.4) for comparison. We see a high interest in studying the possibility of using drones to resupply vehicles or transshipment points for transferring goods.

2.5.2.1 Drone Related Studies

Drone deliveries offer several benefits, mainly that they are autonomous and not impacted by (nor do they impact) road traffic. Drones can fly for a limited time, but they can still travel back and forth in middle-sized cities

Table 2.3: Summary of papers on delivery options. The first two are about location options and the other papers are about transportation modes. The abbreviations: heterogeneous fleet (HF), multiple routes per vehicle (MT), maximum route duration (RD), order pickup range (PR), vehicle capacity (VC), time windows (TW), delivery deadlines (DD), battery charging requirements (BC), single vehicle (SV), and pickup station capacity (PS), vehicle routing (route), scheduling of departure times (schedule).

Literature	Problem	Objective	Uncertainty	Inf. Evolution&Quality	Decision	Constraint
Ulmer & Streng (2019)	vehicle dispatching	min(sum of delivery times)	requests(arrival time), customer pickup time	dynamic & stochastic	schedule; delivery locations	DD; MT; PS; PR; VC
Galiullina et al. (2024)	TSP	min(fulfillment cost)	requests(reaction to incentive)	static & stochastic	incentive; route	SV
Ulmer & Thomas (2018)	VRP	max(number of requests served)	requests(arrival time)	dynamic & stochastic	accept/reject; assignment of services; route	DD; MT; HF; BC
Lin et al. (2018)	VRP	min(sum of transportation cost, fuel cost, and emissions cost)	requests(arrival time, pickup location, delivery location)	dynamic without future knowledge	route	DD; VC
Arslan et al. (2018)	matching and TSP	min(distribution cost: sum of compensation paid to the drivers and cost of dedicated vehicle routes)	requests(arrival time), driver (arrival time)	dynamic without future knowledge	assignment of services; route	TW; VC; RD; HF
Dayarian et al. (2020)	VRP (with drone resupply)	max(percentage of orders served); min(operational costs)	requests(arrival time)	dynamic & stochastic	schedule; route; drone resupply	TW; DD; HF; VC
Pina-Pardo et al. (2021)	TSP (with drone resupply)	min(completion time for serving all customers)	no	static & deterministic	route; drone resupply	HF; MT; VC
Archetti et al. (2021)	VRP	min(distribution cost)	requests(arrival time)	dynamic without future knowledge	route	TW; VC; DD; HF
Chen et al. (2022)	VRP	max(number of requests served)	requests(arrival time, location)	dynamic & stochastic	accept/reject; assignment of services; route	DD; MT; HF; BC
Ausseil et al. (2022)	matching	max(expected revenue)	request (arrival time, pickup location, delivery location), driver (arrival time, choice)	dynamic & stochastic	assignment of services	DD; VC
Liu et al. (2024)	time-expanded flow network	max(profit)	no	static & deterministic	schedule; drone resupply	HF; VC; DD
Pina-Pardo et al. (2024a)	TSP (with drone resupply)	max(#requests served)	requests(arrival time)	dynamic without future knowledge	accept/reject; route; drone resupply	HF; MT; VC
Pina-Pardo et al. (2024b)	VRP (with drone resupply)	min(total completion time)	no	static & deterministic	route; drone resupply	HF; MT; VC
Model (2.3)	OP	min(service time)	no	static & deterministic	schedule; delivery locations	DD; SV; MT; PS
Model (2.4)	OP	min(service time)	no	static & deterministic	schedule; delivery locations; route	DD; SV; MT; PS

conveniently within a 15km radius (Shankland, 2022). When we compare vehicles with drones, we notice that vehicles are capable of delivering many orders at once but often suffer from traffic congestion. At the same time,

the capacity of drones is limited, but they can travel at higher speeds (as they avoid traffic) and with free trajectories. However, drones are sensitive to other factors, such as the wind conditions in landing zones and the density of the population (Gianfelice et al., 2022; Weibel, 2005).

Ulmer & Thomas (2018) present a dynamic vehicle routing problem with a heterogeneous fleet of drones and vehicles. Customers order goods during the day, which must be delivered by the end of the day. The authors propose to use a threshold for dividing the service area into two zones. Then, following the observation that vehicles may be more suitable to serve customers in downtown areas close to the depot while drones may be more appropriate for rural areas with dispersed customers, vehicles are preferred to serve customers in the zone within the threshold. Drones preferably serve customers outside. The authors propose a PFA parameterized over such a threshold. Among a set of potential threshold values, they select the PFA, which leads to the highest average number of customers served within the delivery deadline. Compared to other threshold-type benchmark policies, Ulmer & Thomas (2018) find that districting by the proposed approach is highly profitable as it considerably increases the expected number of requests served. Chen et al. (2022) also choose delivery means between vehicles and drones. Rather than deciding based on customers' locations only (as done in Ulmer & Thomas (2018)), the authors use more information, such as time and fleet availability, to decide whether to use a vehicle or a drone. They propose a deep Q-learning approach to determine the assignment of a new customer to either a vehicle or a drone. The computational results compared to benchmark policies show the effectiveness of the proposed approach.

There is a vast amount of literature on combining drone deliveries with conventional trucks, and the survey on this topic would require an article on its own. We refer interested readers to recent survey articles by Chung et al. (2020), Macrina et al. (2020), and Moshref-Javadi & Winkenbach (2021). Regarding truck and drone routing dealing with online orders in SDD, the number of contributions is rather limited.

Dayarian et al. (2020) propose a home delivery system in which drones are used to resupply trucks for delivery. Trucks follow a predetermined route, while drones are dispatched to deliver new parcels to a truck at a prede-

termined meeting location and time. The authors introduce two resupply strategies: restricted resupply, where trucks are resupplied only after delivering all on-board parcels, and flexible resupply, where trucks can detour to meet drones. The computational results show that the flexible resupply performs slightly better than the restricted resupply. Similarly, Pina-Pardo et al. (2024a) explore real-time decision-making on dynamic arriving customers for truck routing problems with drone resupply. In Pina-Pardo et al. (2024a), they develop an MDP model and adopt different solution approaches. Specifically, they formulate a MILP model for the static counterpart of the problem to derive an upper bound, and then propose an online reoptimization-based policy. They show that their average fill rate achieved by the proposed policy is at most 20% lower than the perfect information model, and the computational times to make a decision are hundredths of a second.

Pina-Pardo et al. (2021) study a similar problem using drones to resupply trucks, as done by Dayarian et al. (2020) and Pina-Pardo et al. (2024a). They assume that release dates and locations of orders are known in advance. They model the problem as a TSP with release dates and drone resupply. First, they provide a MILP formulation to minimize the time needed to deliver all orders. Then, they propose a two-stage decomposition heuristic. The experiments compare the system to a traditional truck-only delivery system and show that drone resupply can reduce the total delivery time. Building on this, Pina-Pardo et al. (2024b) extend the work by allowing multiple returns of both trucks and drones to the depot. Unlike the earlier studies, drones in this model can resupply any truck, adding more flexibility. They develop a matheuristic approach to solve both the truck-and-drone and truck-only versions. Computational results show that drone resupply increases fill rates by 21% compared to the truck-only version. Liu et al. (2024) also consider multiple trucks and drones, but they focus on cases where drones send packages to trucks on the route, and then trucks make the final deliveries. Specifically, the trucks follow predefined routes but may wait at specific locations to receive parcels from drones. The goal is to maximize delivery profit within a time horizon. The authors formulate the problem as a time-expanded network flow model and propose a greedy algorithm. Their results show the optimality gap between their greedy approach and the optimal solution (obtained by solving an ILP formulation) is within 10%.

2.5.2.2 Crowdshipping Related Studies

The concept of crowdshipping is based on incentivizing private citizens to deliver packages by offering a compensation. This strategy is widely deployed nowadays (see Walmart (Barr & Wohl, 2013) and Amazon (Bensinger, 2015) for early implementations of the idea). Ulmer & Savelsbergh (2020) and Dai & Liu (2020) investigate workforce scheduling and capacity planning of crowdsourced drivers, respectively, to achieve a satisfying service level at a minimum cost. Dayarian & Savelsbergh (2020) quantify the possible benefits of crowdshipping for SDD and demonstrate the value of future information in a highly dynamic and stochastic SDD environment.

Routing problems with crowdshipping are also called *routing problems with occasional drivers*. This term was introduced in Archetti et al. (2016). In the context of SDD, we highlight several contributions. Arslan et al. (2018) investigate a problem in which a provider receives orders online and matches them to ad hoc drivers or dedicated vehicles for which the associated routes are also determined. Orders and drivers arrive dynamically over time. The objective is to minimize the sum of the compensation paid to the crowdsourced drivers and the cost of the dedicated vehicle routes to serve all customers. To solve the matching problem, the authors propose a rolling horizon framework and develop an exact solution method that is applied whenever there is new information. Archetti et al. (2021) also consider a setting where customer orders arrive online. The company managing deliveries has a set of "regular" drivers (drivers working regularly for the company) and crowdsourced drivers (occasional drivers). There are four major differences compared to the study of Arslan et al. (2018): the information related to occasional drivers is deterministic; not all parcels need to be delivered on the same day, all parcels must be dispatched from the same depot, and route diversions are allowed. The objective is to serve all customers at a minimum cost. Besides the routing cost for regular drivers and the compensation cost for occasional drivers, the authors also introduce penalties for violating time windows or postponing the delivery to the next day. An iterated local search is proposed to solve the problem, outperforming an insertion heuristic.

When comparing different delivery paradigms, Lin et al. (2018) consider three options: hub-and-spoke, SDD

with a commercial fleet and SDD with crowdshipping. In the first one, all requests are collectively received by a hub-and-spoke delivery system by the end of the day and scheduled for next-day service. In the two SDD paradigms, requests are added in real-time and are served during the same working day. The results find hub-and-spoke to be cost-effective for the traditional delivery service but expensive for the SDD service. They also find that crowdshipping provides a low-cost solution to SDD.

Finally, Ausseil et al. (2022) consider a peer-to-peer logistics platform to dynamically match crowdsourced occasional drivers with customer requests. Contrary to what is assumed in the abovementioned contributions, crowdsourced drivers might not accept serving a customer. In order to mitigate acceptance uncertainty, the platform offers each driver a menu of requests to choose from. The objective is to maximize the expected revenue, i.e., the platform rewards obtained from successfully assigning orders to drivers within the time horizon. To treat the menu selections, the authors present a multiple-scenario approach, including repeatedly sampling potential driver selections, solving two-stage stochastic models, and deriving solutions through a consensus algorithm.

2.5.3 Common Findings and Gaps

The literature suggests that incorporating diverse transportation modes can improve the adaptability and resilience of SDD networks. While this progress brings benefits, it also introduces new uncertainties related to unfavorable weather conditions (particularly relevant for drones), the available capacity of shared pickup stations, and the availability of crowdsourced drivers. There has been a trending research shift toward dynamic and real-time optimization, incorporating elements such as drone resupply and flexible delivery locations. However, many studies tend to overlook these uncertainties, with the sole source of uncertainty considered being the release dates. Besides, the solution approaches applied in these works mostly rely on well-established methods such as reoptimization, approximate dynamic programming, and scenario sampling. While these approaches have proven effective, it remains open if they can efficiently address a broader range of uncertainties.

2.6 Pickup and delivery

This section introduces same-day pickup and delivery problems. We define these problems, followed by a literature review organized into subsections. Section 2.6.1 covers general same-day pickup and delivery issues, while Sections 2.6.2 and 2.6.3 focus on two typical applications: meal delivery and personal shopper services. We conclude this section by summarizing common findings and identifying gaps in the literature.

When parcels are associated with origins and destinations, the problems are known as pickup and delivery problems (Savelsbergh & Sol, 1995). The papers are summarized in Table 2.4. We observe that performance measures are more varied than those in previous sections. Additionally, new uncertainty factors arise, such as driver parking time and food ready time. Furthermore, there is a higher diversity in solution approaches.

Table 2.4: Summary of papers on pickup and delivery. The first five papers are on general pickup and delivery problems whereas the remaining papers discuss two applications - meal delivery services and personal shopper services. The abbreviations: multiple routes per vehicle (MT), vehicle capacity (VC), time windows (TW), and delivery deadlines (DD), vehicle routing (route), scheduling of departure times (schedule).

Literature	Problem	Objective	Uncertainty	Inf. Evolution&Quality	Decision	Constraint
Mitrović-Minić & Laporte (2004)	VRP	min(travel distance)	requests(arrival time, pickup & delivery locations)	dynamic without future knowledge	schedule; route	TW; DD
Pureza & Laporte (2008)	VRP	min(#lost sales, routes, travel distance)	travel time, requests(arrival time)	dynamic without future knowledge	accept/reject; route; schedule	TW; DD; VC
Ghiani et al. (2009)	VRP	min(customer inconvenience)	requests(arrival time)	dynamic & stochastic	schedule; route	TW; DD;
Ghiani et al. (2022)	VRP	min(customer inconvenience)	requests(arrival time)	dynamic without future knowledge	schedule; route	TW; DD
Ackva & Ulmer (2024)	VRP	max(#requests served)	requests(release time, pickup store, location)	dynamic & stochastic	route; schedule	TW; DD; VC
Yildiz & Savelsbergh (2019a)	VRP	optimize performance measures (courier compensation, click-to-door time...)	no	static & deterministic	assignment of services; route; schedule	DD; MT
Mao et al. (2019)	VRP	no	no	static & deterministic	assignment of services	TW
Ulmer et al. (2021)	VRP	min(sum of delay)	requests(arrival time, ready time)	dynamic & stochastic	route; assignment of services	TW; DD; MT
Hildebrandt & Ulmer (2021)	VRP	max(#requests served)	requests(restaurant choice, preparation time), driver (parking and delivery times)	dynamic & stochastic	route	TW; DD
Arslan et al. (2021)	VRP	max(#requests served)	requests(arrival time)	dynamic without future knowledge	accept/reject; route	TW; DD; VC
Auad et al. (2023)	matching	balance courier and customer satisfaction	requests(arrival time, ready time), delivery capacity (courier availability)	dynamic without future knowledge	assignment of services	TW; DD
Neria et al. (2024)	VRP	min(sum of delay)	requests(arrival time, delivery location, meal preparation time, food type)	dynamic & stochastic	schedule; route; assignment of services	DD; VC; MT

2.6.1 Literature review on same-day pickup and delivery problems

One of the first contributions of studying a same-day pickup and delivery problem (SDPD) is due to Mitrović-Minić & Laporte (2004). The authors focus on same-day pickup and delivery of letters and small parcels by inves-

tigating the impact of deploying four possible waiting delivery strategies. Pureza & Laporte (2008) study a similar problem as Mitrović-Minić & Laporte (2004), but they allow rejection of requests. Also, the distances are time-dependent, i.e., they may vary according to the time of the day.

Another early contribution is due to Ghiani et al. (2009). The authors investigate a multi-vehicle dispatching problem with pickup and delivery where requests arrive according to a known stochastic process. The goal is to minimize the expected total customer inconvenience, where inconvenience for each customer is measured by the delivery time exceeding their own soft deadline. A sampling technique used to anticipate future demands is embedded in the cheapest insertion heuristic and a local search algorithm to design the delivery routes. Their experiments show that the algorithm achieves up to 69% improvements over a pure reactive procedure. More recently, Ghiani et al. (2022) study a similar dynamic pickup and delivery problem, but they do not assume any explicit probabilistic knowledge of the request arrival process. They propose a parametric PFA in which a supervised learning model selects the best parameter setting online. Their numerical study shows that the policy is scalable to handle instances of increasing size and provides better performance than two reactive policies.

Lin et al. (2018) and Arslan et al. (2018) (introduced in Section 2.5.2) also address SDPD. When comparing three delivery paradigms, Lin et al. (2018) also consider delivery requests associated with a pair of pickup and delivery locations. Similarly, when matching orders with ad-hoc drivers and dedicated vehicles, Arslan et al. (2018) allow orders to be associated with different pickup locations (a warehouse, a dedicated pickup point, or a retail shop).

Ackva & Ulmer (2024) introduce a pickup and delivery problem with micro-hubs. Each parcel's pickup and delivery locations are mapped to their nearest micro-hubs. This setup allows vehicles to consolidate parcels from different locations for joint delivery. Vehicles operate on a consistent, predetermined daily schedule, visiting micro-hubs at a fixed sequence and time. The objective is to find a consistent routing schedule, i.e., visiting the same micro-hubs at the same times every day, that maximizes the expected number of parcels delivered daily. The authors

model the problem as a two-stage stochastic program: the first stage determines a sequence of stops at micro-hubs with departure times; the second stage decides which parcels to serve, given the schedule. The problem is solved using scenario sampling as introduced by Bent & Van Hentenryck (2004), and four problem-specific consensus functions (CF) are proposed. The computational results show that their approach outperforms the original CF proposed by Bent & Van Hentenryck (2004) and two benchmark heuristics.

2.6.2 Literature review on meal delivery service

Restaurant meal delivery is a typical application for same-day/instant pickup and delivery problems. Fierce competition in this sector requires efficient scheduling and routing optimization algorithms to gain a competitive advantage. We distinguish between the *order time* and *ready time* in these applications. Order time corresponds to when a customer sends a request, choosing from various restaurants. Ready time is when the ordered meal becomes available at the restaurant. The customers are usually unknown until orders are placed, and the ready time of food is also unknown, which creates tremendous challenges in managing the delivery service. Once a request is sent, the order is passed to a delivery company, which matches it to a courier for picking up and delivering. The delivery company has to anticipate future uncertain requests in terms of location and timing, as well as the ready time and delivery routes, before deciding how to match orders with the couriers.

Yildiz & Savelsbergh (2019a) develop an exact solution approach for the meal delivery problem introduced by Reyes et al. (2018a). The authors assume perfect information concerning the order arrival stream, including order request time, order placement time, release date, and the drop-off location. Ulmer et al. (2021) study a similar problem setting, but they assume certain probability distributions of order times, ready times, and customer locations. In such a stochastic and dynamic problem, the major challenge is to have accurate estimations of ready times to prevent the courier from waiting and, thus, potential delays along the route. The customers are promised that the orders will be delivered by a specific deadline, and the overall objective is to minimize the expected delays.

The authors model the problem as an MDP and propose an anticipatory customer assignment policy with a time buffer as a type of CFA. The time buffer is a "cost" added to the scheduled arrival times at the customers to account for the stochasticity. The policy postpones the assignment of a courier to an order to get more information on ready time and future orders. The postponement enables the potential bundling, i.e., assigning multiple orders simultaneously to the same courier. In the computational study, Ulmer et al. (2021) demonstrate the importance of the proposed time buffer compared to solely using postponement and bundling.

While the studies mentioned above assume fixed customer choices, they do not consider the possible impact of late deliveries on customer satisfaction. Mao et al. (2019) examine the importance of on-time and early delivery empirically based on two-month transactional data from an online meal delivery platform in China. They find that the negative impact on customer satisfaction due to late deliveries is much more severe than the positive impact of early deliveries.

Hildebrandt & Ulmer (2021) also work on meal pickup and delivery problems. A novelty of the paper lies in considering that customer choice is affected by estimated arrival time. That is to say, customers can choose between different products and services, and their choice is based on the restaurant preference and the expected delivery time. Thus, the authors work on estimating meal arrival times to inform customers at the time they make their selection. Two supervised learning approaches for the prediction of meal arrival times are proposed. The results show that accurate arrival times can raise service perception and guide customers' selection to improve the delivery system. Besides, the authors find that arrival time predictions help reduce delays but may block customers from ordering. The studies above involve meal preparation from decentralized restaurants. More recently, Neria et al. (2024) investigate meal pickup from ghost kitchens—centralized facilities where there are multiple restaurants preparing online orders for delivery via a shared fleet. To ensure fast delivery and freshness, the study works on dynamic scheduling of food preparation and delivery. Given the complex decision space, the authors propose a large neighborhood search algorithm where decisions are evaluated via a VFA. Their tests indicate that guaranteeing freshness reduces flexibility in food preparation and requires more direct trips.

Besides satisfying customers, courier satisfaction is another aspect to be considered. The works mentioned above assume that couriers do not make autonomous decisions. In fact, couriers may reject certain deliveries due to long distances. Logistics companies typically compete for the same set of couriers who can subscribe to multiple platforms. Thus, it is necessary to maintain courier satisfaction as well. Auad et al. (2023) consider a pickup and delivery problem by combining customer-centric and courier-centric performance metrics. Assuming couriers prefer to operate in relatively small geographic areas, they propose an optimization-based rolling horizon algorithm for managing couriers by coping with region resizing and delivery task assignment decisions. Their empirical study using real-world data shows that for a fixed courier fleet size, courier satisfaction metrics can be increased without sacrificing key performance metrics related to customer service.

2.6.3 Literature review on personal shopper service

Personal shopper service experienced a significant growth in online grocery (Nesin, 2022). Compared to delivering fresh meals, there is more time flexibility here. However, the personal shopper service often requires visiting multiple stores to serve a single customer rather than visiting one pickup location per customer. Thus, the problems can be seen as an extension of pickup and delivery problems with multiple pickup locations per order.

Pickup and delivery problems in on-demand personal shopper service are investigated by Arslan et al. (2021). In a classical pickup and delivery setting, the pickup time is often negligible compared to travel times. However, in personal shopper service applications, the pickup time cannot be ignored, as it increases with the number of pickups per store. Arslan et al. (2021) aim to maximize the number of requests served, subject to fixed shopper capacity (the number of tasks a shopper can do). The authors propose three different order-consolidation strategies. Their results show that order consolidation, especially shopping consolidation, can increase the number of requests served within the available shopper capacity.

2.6.4 Common Findings and Gaps

Unlike previous problems, where all items are collected from a common depot, pickup and delivery problems introduce additional complexities, such as multiple pickup locations and/or differing processing times. Strategies for handling the consolidation challenge range from simple heuristics, like waiting at the current location to batch multiple requests, to more complex approaches involving estimations of future requests or food ready time with ML techniques. From the business perspective, there remain many features to explore, such as a dynamic pricing mechanism to incentivize customers to choose a longer delivery-time option during peak hours, which helps to balance demand and reduce pressure on the delivery network.

2.7 Districting - fleet size and mix

This section reviews the literature on districting problems. We start by providing background information on this topic. Then, in Section 2.7.1, we discuss general tactical design decisions that arise in a "standard" SDD system, wherein orders placed by a cutoff time are delivered by vehicles that return to the depot by the deadline. In Section 2.7.2, we present new variants arising in SDD associated with various features, such as fairness, hub capacity, and more. We conclude with common findings and identify some gaps in the existing literature.

Decisions at distribution companies are carried out at three levels (see, e.g., Jabali et al. (2012)). At the *strategic* level, decisions are made concerning the types and numbers of vehicles that need to be purchased to fulfill the expected demand. At the *tactical* level, leasing additional capacity or subcontracting part of the fleet might be needed. Finally, at the *operational* level, daily operations, which usually concern the routing of vehicles, are conducted.

So far, we focused on operational problems. However, other relevant problems at the strategic and tactical levels need to be considered in the context of SDD. One of the first optimization problems integrating strategic

and tactical decisions in the general routing context is introduced by Golden et al. (1984). Their *fleet size and mix vehicle routing problem* asks to determine the optimal number of heterogeneous vehicles with possibly different costs and capacity to lease or purchase. The authors consider the problem of minimizing the cost, including vehicle-dependent fixed cost and variable cost components such as fuel, maintenance, and manpower. Over the years, more papers have appeared on the standard fleet size and mix VRP and its variants. We refer readers to the survey by Koç et al. (2016) that classifies and reviews the contributions on this topic.

In SDD, a single depot tends to serve a relatively large region with hundreds or even thousands of parcels delivered daily. Hence, it is highly impractical to assume that each vehicle should be able to serve the whole region. Instead, customer locations are partitioned into zones (districts) with dedicated vehicles, where each district can be served by either a single vehicle (Banerjee et al., 2022) or a small fleet (Stroh et al., 2022). Such partition of a service region is also known as *districting*. Thus, the districting problems belong to the strategic or tactical decision-making levels (Haugland et al., 2007). The papers are summarized in Table 2.5, including the first paper on crowd-source meal delivery and the next two on general tactical decisions of SDD systems, followed by a paper on issues related to SDD service times and regions, and the last on service network design with hub capacity.

Table 2.5: Summary of papers on districting. The abbreviations: single vehicle (SV), multiple routes per vehicle (MT), multiple depots (MD), vehicle capacity (VC), time windows (TW), and delivery deadlines (DD).

Literature	Problem	Objective	Uncertainty	Inf. Evolution&Quality	Decision	Constraint
Yildiz & Savelsbergh (2019b)	service and capacity planning in on-demand meal delivery platforms	max(profit of an online meal delivery platform)	requests(arrival time), mean delivery time, courier(arrival time), delivery offer acceptance-probability for couriers	static & stochastic	delivery capacity and composition, delivery service charge, courier compensation, service coverage area	DD
Stroh et al. (2022)	tactical design of SDD systems	min(delivery time)	requests(arrival time, location)	static & stochastic	fleet sizing, order cutoff time, and consolidation with overnight orders	DD; VC; MT; SV
Banerjee et al. (2022)	tactical design of SDD systems	min(#vehicles to serve the region)	requests(arrival time)	static & stochastic	fleet sizing, service region partitioning	DD; MT
Banerjee et al. (2023)	tactical design of SDD systems	max(#requests served)	requests(arrival time, location)	dynamic & stochastic	order cutoff time, service coverage area	DD; MT
Wu et al. (2023)	service network design for SDD system	min(cost of a vehicle schedule that maximizes #commodities for which a feasible path can be found)	no	static & deterministic	route; schedule	TW; DD; VC; MD

2.7.1 Literature review on general tactical design decisions

Stroh et al. (2022) propose a vehicle-dispatching model for answering some tactical design questions in SDD distribution. Their model uses continuous approximation techniques to capture the average behavior of an SDD system. Specifically, they consider deliveries from a single distribution center, and major decisions are concerned with selecting the optimal fleet size, determining an order cutoff time, and combining SDD with overnight order delivery operations. The objective is to minimize the total delivery time spent by all vehicles serving orders received within a cutoff time. The computational study using realistic data shows that the model is good at predicting the system's behavior. Banerjee et al. (2022) also use continuous approximations to capture average-case operational behavior. However, rather than assuming every vehicle may serve orders from the whole region, as in Stroh et al. (2022), this paper focuses on deciding the minimum number of vehicles to serve a service region when it can be partitioned into zones dedicated to individual vehicles. The proposed method first maximizes the area associated with a single-vehicle delivery zone independently and then determines optimal areas as a function of the distance

from the depot. As for the first step, the authors propose a Voronoi approach to partition the service region into single-vehicle zones. The Voronoi approach creates zones by assigning each generator location in a Euclidean plane the set of delivery locations closer to it than any other generator location. Afterward, the fleet size is estimated based on the surface of the service area. The simulations suggest that combining the proposed simple operational dispatching policy with the Voronoi partitioning produces solutions that meet order cutoff time in most cases.

2.7.2 Literature review on new variants

Based on traditional logistic districting problems focusing on fleet sizing and routing, several new variants associated with, e.g., planning crowdsourced delivery capacity, customer fairness, fleet assignment with heterogeneous fleets, assignment of couriers, and hub capacity, appeared in the context of SDD. In the following, we elaborate on these variants and present research works dedicated to each.

Planning crowdsourced delivery capacity: Yildiz & Savelsbergh (2019b) investigate service and capacity planning problems in crowdsourced meal delivery. Unlike Stroh et al. (2022) and Banerjee et al. (2022), who focus on fleet size optimization or maximization of vehicle utilization for an SDD system, their model aims to maximize the profit of an online meal delivery platform while ensuring a target service quality level. They analyze factors like service area coverage, courier compensation, probabilities of courier order acceptance, and composition of delivery capacity (crowd-sourced couriers and company-employed drivers). The analysis with numerical examples offers several interesting insights, including the finding that the hybrid delivery capacity (crowdsourced couriers and company-employed drivers) can improve reliability and profit.

Customer fairness: The work of Banerjee et al. (2023) contributes to the fairness literature. Unlike the papers (Stroh et al., 2022; Banerjee et al., 2022) mentioned above, which assume that orders arrive from a predefined service region before a cutoff time, Banerjee et al. (2023) consider the service region as a decision variable and allow it to

vary throughout the day. The aim is to select a service region and deadline that maximizes the expected number of orders served. They offer various order cutoff times to different areas. The authors find that variable-area systems yield higher revenue. However, fixed-area systems provide more fairness to customers within the systems.

Fleet assignment with heterogeneous fleets: Another possibility to be effective in SDD services is to partition the service region according to the geographical location of requests and decide the delivery mode for each subregion according to the viability of each mode, as we have seen in Section 2.5.2. For example, Ulmer & Thomas (2018), whose work was introduced in Section 2.5.2, propose to divide the service area into two zones, one of which is served by drones and the other by vehicles. Their study focuses on an operational problem where fleet assignment decisions are made dynamically throughout a day using a threshold-based PFA.

Assignment of couriers on-demand: In a rapid delivery system, the assignment of couriers to demand points is a crucial decision (Agatz et al., 2012; Savelsbergh & Van Woensel, 2016; Archetti & Bertazzi, 2021). The work of Auad et al. (2023), introduced in Section 2.6.2, proposes a dynamic couriers assignment policy for SDD. Initially, small operating regions are allocated to couriers, but these boundaries can be dynamically expanded based on demand. That way, flexibility in assignment decisions is introduced, to increase couriers' working efficiency. Couriers' satisfaction is improved, as they are no longer expected to wander over the full-service region. The computational experiments show that the proposed strategy increases courier satisfaction with a minor impact on customer satisfaction.

Hub capacity: Hubs are facilities that switch, sort, connect, and consolidate/disaggregate traffic between multiple origins and destinations (Alumur et al., 2021). Limited hub capacities restrict the number of vehicles that can simultaneously be loaded and unloaded in a hub. In urban SDD, with an aggressive amount of orders, hub capacity constraints can lead to waiting at hubs, which, in turn, renders specific shipments infeasible. Wu et al. (2023) investigate the problem with hub capacity constraints. The goal is to determine a routing path for each commodity so that the loading and unloading times, together with capacities at hubs, are respected. They consider

a lexicographic objective function that first maximizes the number of commodities served and then minimizes the total travel distance. They model the problem on a time-expanded network, commonly done when time is critical (Belieres et al., 2021). Each time-expanded node stands for a stakeholder, i.e., supplier, warehouse, or customer, at a given period. The problem is formulated as a MILP, and three heuristics are proposed to solve real-world instances efficiently. This is one of the few problems that tackle tactical decision problems in SDD, different from districting.

2.7.3 Common findings and gaps

Research on districting in SDD has focused on tactical models that capture average-case operational behavior. Recent studies have expanded this by exploring various variants, such as customer fairness and dynamic districting, which allow for more flexible and responsive SDD operations. However, challenges remain in integrating these models to address multiple competing objectives, such as balancing customer fairness, profitability, and the number of customers served across different service areas. Future research could explore the multi-objective optimization frameworks that balance these factors.

2.8 Model comparison

This section analyzes structural differences of optimal solutions obtained when solving formulations (2.1)-(2.4). The analysis of solutions and illustrative examples given in the following serve several goals:

- Concept clarification: they help elucidate and clarify the underlying concepts associated with the problems studied in the literature.
- Parameter sensitivity: they help validate if specific parameter values can exert a notable influence on the

outcomes.

- Impact of objectives and constraints: they help illustrate how solutions change when different constraints or objectives are imposed.

We compare the obtained solutions in terms of customer fairness, quality of service, and possible environmental impacts. For each problem model, we list metrics related to service quality, fairness, and environmental impact. Specifically, travel distance measures environmental impact, service rate and average waiting time measure service level, and max waiting time and waiting time variability (the difference between max waiting time and min waiting time) measure fairness. Besides, except for the service rate, all the aforementioned metrics are calculated based on the requests served.

Data about the first instance and its optimal solutions associated with each of the four formulations are presented in Figure 2.5. The instance has eight orders to be delivered within the time horizon [0, 250]. Each order is associated with a release date valued from 0 to 123, which remains the same over all formulations. As for the time slot option problem (Formulation (2.2)), there are three SDD options, i.e., $|\Delta| = 3$ with deadlines $D_\delta \in \{60, 120, 240\}$ minutes. Willingness To Pay for each customer and each SDD option is provided in the third row of the table shown in Figure 2.5. For example, the customer requesting order 1 is willing to pay at most 40 to get the order delivered within 60 minutes. Concerning the problem with delivery options (Formulation (2.3)), there are three stations with a storing capacity of 3 orders. We determine the set of stations accepted by each customer according to the travel time radius (30 time units) away from the customer's home. The last line of the top left table lists the potential stations for delivering orders. For instance, order 1 can be delivered to station 1 or 3 as long as there is available capacity. The coordinates of points are denoted in brackets in the squares and we use Euclidean distances. Circles represent customer locations, whereas rectangles correspond to pickup locations. The central depot is shown as a double circle. Finally, we fix the value of M to 250 in formulations (2.3) and (2.4).

The bottom part of Figure 2.5 represents, from left to right ((a) to (d)), the optimal solutions obtained by

Order	1	2	3	4	5	6	7	8
Release Date/ Order Time	0	15	19	28	45	80	110	123
WTP for options 60, 120, 240(min)	40, 20, 10	80, 40, 20	40, 20, 10	80, 40, 20	40, 20, 10	80, 40, 20	40, 20, 10	80, 40, 20
Stations	1, 3	2	3	3	1	1, 3	2, 3	3

Metric \ Model	Release Date	Time Slot	Direct Trip	Routing
travel distance	234	243	196	190
service rate	88%	75%	100%	100%
avg waiting time	107	76	69	68
max waiting time	175	175	149	149
waiting time variability	159	152	126	126

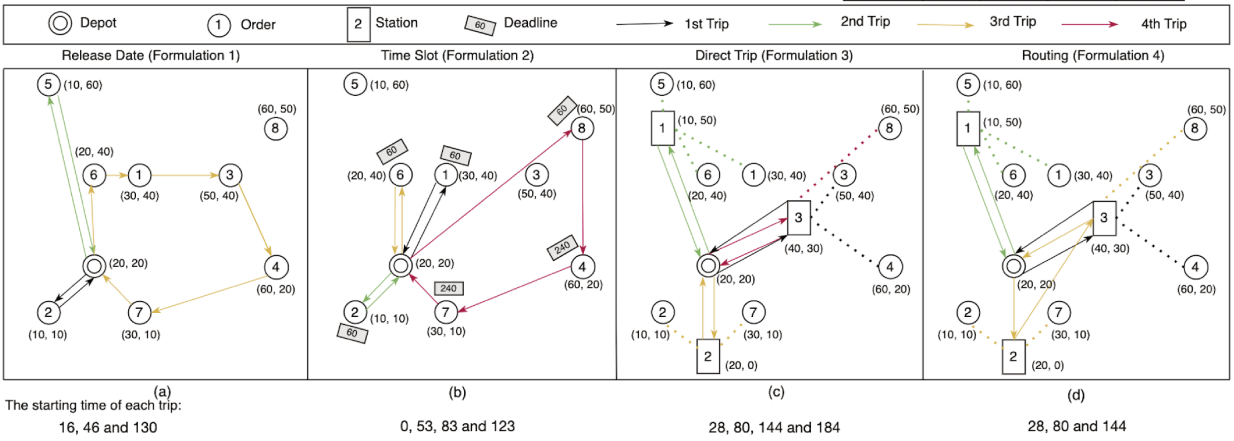
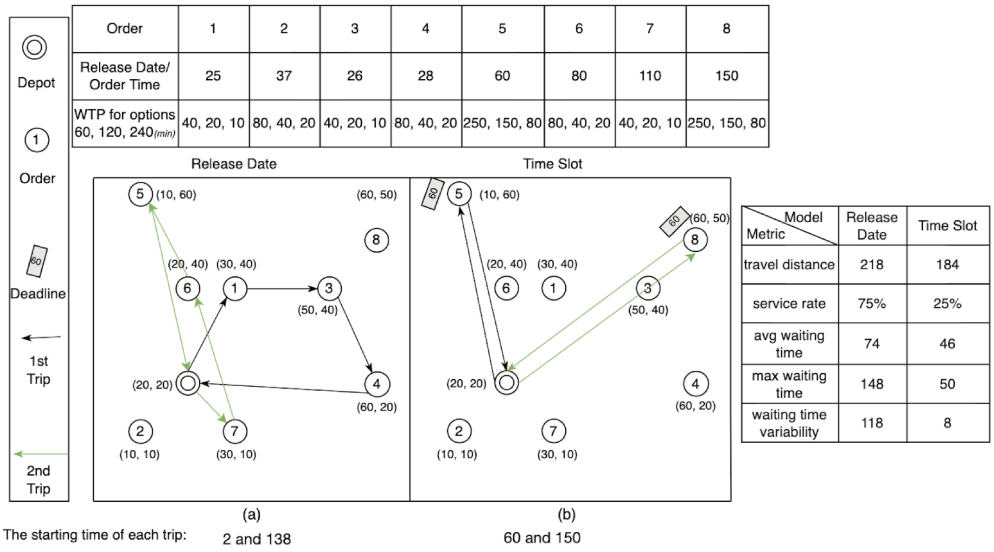


Figure 2.5: Small instance and corresponding solutions. The starting time of each trip in each block from left to right in order: 16, 46 and 130; 0, 53, 83 and 123; 28, 80, 144 and 184; 28, 80 and 144

solving formulations (2.1)–(2.4), respectively. Figure 2.5(a) shows the solution of the problem with release dates (Formulation (2.1)). We observe that seven requests can be served by three trips before the deadline. Comparing it with Figure 2.5(b), the solution of the time slot problem (Formulation (2.2)), we notice that order 8 instead of orders 3 and 5 is served, which is due to the fact that order 8 has a higher WTP which leads to a higher profit. This example illustrates that due to the maximization of the expected revenue, not only fewer parcels may be served within the same deadline, but also more trips may be needed. Indeed, one more trip is dispatched for the solution of Formulation (2.2) compared to the solution of Formulation (2.1). In the time slot problem, if there are a few late customer orders with very high WTP located very far from the depot, it might be that the solution serves those few customers and leaves many other customers unserved. This is illustrated in Figure 2.6(b) where the WTP of orders 5 and 8 has been increased to (250, 150, 80) and their release dates have been increased from 45 and 123 to 60 and 150 minutes, respectively. We see that in this case customers 5 and 8 only are served. The solution is compared with the one of Formulation (2.1) (with release dates, depicted in Figure 2.6(a)), where six requests are served instead. This highlights the fact that the time slot model might disappoint certain customers. This is detrimental to retaining customers' loyalty in a highly competitive environment. Furthermore, the pursuit of maximum profit may obstruct consolidation opportunities and lead to a higher frequency of trips, causing negative externalities on the environment, which can be seen in the upper-right table of Figure 2.5: while the time slot model serves the fewest customers, it incurs the greatest travel distance. A possible improvement on the model can be achieved by introducing two hierarchical objectives: maximizing the number of customers served and maximizing the revenue.

Now let us turn our attention to the problems with delivery options. Formulation (2.3) models the case where only direct trips are allowed to replenish pickup stations. In the solution shown in Figure 2.5(c), four trips serve all requests with a service time of 553. On the other hand, when enabling visiting more than one pickup station in a trip (Formulation (2.4)), only three trips are needed to serve all requests, and the service time is reduced to 547. The upper-right table of Figure 2.5 also shows both the direct trip (Formulation (2.3)) and routing models (Formulation (2.4)) get shorter travel distances and reduced average and maximum waiting time when compared

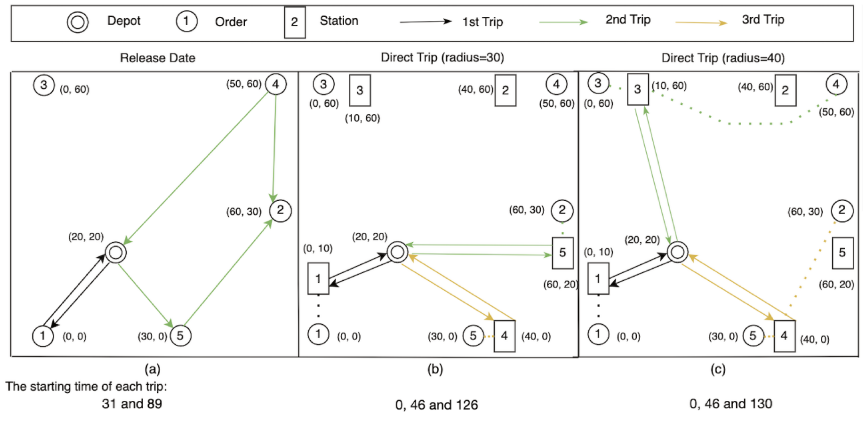


Order	1	2	3	4	5	6	7	8
Release Date/ Order Time	25	37	26	28	60	80	110	150
WTP for options 60, 120, 240(min)	40, 20, 10	80, 40, 20	40, 20, 10	80, 40, 20	250, 150, 80	80, 40, 20	40, 20, 10	250, 150, 80

Metric \ Model	Release Date	Time Slot
travel distance	218	184
service rate	75%	25%
avg waiting time	74	46
max waiting time	148	50
waiting time variability	118	8

Figure 2.6: New instance and solutions obtained after changing the WTP and release dates of orders 5 and 8. The starting time of each trip in order: 32 and 138; 60 and 150

to the previous two models, which shows their better performance with respect to environmental impact, fairness, and service rate. Furthermore, the routing model achieves a slightly shorter travel distance and less average waiting time when compared to the direct trip model. Nevertheless, direct trip models can still be interesting, if eco-friendly delivery alternatives are used, such as droids or drones (Ulmer & Streng, 2019). Indeed, it is often assumed that these two means of transportation can handle one package at a time. An extreme situation arises when there is only a single feasible pickup station per each customer. Figure 2.7 illustrates such a case: there are five orders with release dates ranging from 0 to 80 to be delivered before the deadline $T_E = 250$. In the table of Figure 2.7, the column named *Stations 1* lists the feasible pickup stations for formulations (2.3) and (2.4) with radius equal to 30, and *Stations 2* lists the ones with a bigger coverage, i.e., radius equal to 40. The capacity of each station is 2. The solution shown in Figure 2.7(b) is obtained by setting the radius equal to 30, in which case, each pickup station can serve one customer only. We notice that only three customers can be served before the deadline, and the service time is 586. When increasing the radius from 30 to 40, we observe that all customers can be served (see Figure 2.7(c)), and the objective value is reduced to 339. In the lower-right table of Figure 2.7, we see that,



Order	Release Date/Order Time	Feasible pickup stations(radius=30)	Feasible pickup stations(radius=40)
1	0	1	1, 4
2	25	5	4, 5
3	28	3	2, 3
4	45	2	2, 3
5	80	4	1, 4

Metric \ Model	Release Date	Direct Trip(radius=30)	Direct Trip(radius=40)
travel distance	206	184	188
service rate	80%	60%	100%
avg waiting time	104	53	68
max waiting time	155	75	134
waiting time variability	110	52	111

Figure 2.7: New instance and solutions obtained when changing the radius for determining feasible pickup stations. The starting time of each trip in each block from left to right in order: 31 and 89; 0, 46 and 126; 0, 46 and 130

in comparison to the previous model, the direct trip model with a bigger coverage improves service rate from 60% to 100% with a marginal increase in travel distance and average waiting time but with a bigger increment of max waiting time and waiting time variability. By comparing the solution obtained using pickup stations with the one obtained using the release dates model (see Figure 2.7(a)), we observe that pickup stations may be advantageous in terms of the number of requests served and the total service time. However, this is not always the case and it highly depends on the location of pickup stations and the radius customers are willing to move in order to get their package. Thus, these two factors have to be carefully considered during strategic and tactical design phases, before deploying pickup-station-based models.

To conclude, we observe that changing constraints and objectives can lead to different routes and order assignments for the same instance. To select a fair and robust model, a sensitivity analysis including extreme case instances may help in better understanding the pros and cons of the proposed models. A deliberate selection of objectives, constraints, and broad tests to fulfill both the customer needs and company expectations, while keeping sustainability goals in mind, is imperative.

2.9 Managerial insights and future research directions

Despite the continuous development of new delivery technologies and advances in analytical methods, providing satisfactory services to customers while making positive profits is still challenging in SDD services. Besides, the environmental impacts associated with SDD operations are negligible. We summarize in the following the most challenging topics that are partially covered in the SDD problems reviewed in this survey and those that still need to be addressed by future research.

Increased uncertainties coupled with tight delivery times: There are several sources of uncertainty in SDD, such as unknown customer requests arriving during the day, uncertain customer locations and demand, traffic jams during peak hours, or unexpected road works that may require rerouting. Some of these sources of uncertainty affect other distribution problems as well. However, in SDD, they have a stronger impact due to the very tight delivery times. In routing problems with release dates and deadlines (see Section 2.3), time uncertainty is one of the major obstacles to consolidation. For home-attended deliveries (see Section 2.5), to avoid missing the delivery deadline, logistic companies often propose to customers a set of delivery time slots or different delivery options. However, this renders the corresponding optimization problems even more challenging to solve. Hildebrandt et al. (2023) highlight two challenges in solving such stochastic and dynamic problems - searching for a solution in vast solution space and evaluating each solution considering future information changes and actions. In this article, we have outlined recent approaches to cope with uncertainties in order to exploit consolidation opportunities better or increase the number of orders delivered on time. Exploiting historical data to predict release dates or order times, combined with anticipative algorithms, is a widely used approach to avoid myopic behaviors. In practice, managers can consider building operational buffers, for instance, maintaining a flexible workforce, deploying extra vehicles during peak periods, or using reoptimization approaches that allow for real-time adjustments. Another strategy is to assign couriers to small operating regions, with the flexibility to expand these boundaries on demand as conditions change (see Auad et al. (2023)). However, the research in this area is still limited, and we believe it

will grow along with the increasing coverage of SDD services in the future.

Runtime of online algorithms: In SDD, algorithms developed for operational decisions must be fast enough to ensure prompt decision-making. In decisions involving direct customer interactions (through online platforms, for example), results must be provided within (milli) seconds, thereby avoiding disappointing customers; in vehicle dispatching, decisions may be made within minutes to enable seamless deliveries. Most literature we reviewed underscores the importance of adhering to strict runtime constraints (e.g., Mitrović-Minić & Laporte (2004); Van Heeswijk et al. (2019); Klapp et al. (2020); Li et al. (2024b)). However, with increased geographic scales and more customized options, such as multiple delivery options and specific customer preferences, problem settings are becoming more complex, and researchers face challenges in developing new algorithms to keep pace with the rising complexity of problems.

At a managerial level, they can implement flexible routing solutions that quickly adjust to new delivery requests or traffic delays. To this end, routing systems need to be connected with advanced tools that process real-time data to ensure timely and efficient route adjustments. At the same time, while advanced dynamic and stochastic solution approaches can increase SDD performance, smaller companies must assess their return on investment carefully. In scenarios with a lower volume of deliveries, simpler and more cost-effective deterministic approaches may offer sufficient values. Managers should conduct a cost-benefit analysis to decide the most appropriate level of technology adoption, ensuring efficient resource allocation based on their operation scale.

Autonomous vehicles and drones: For safety reasons, strict environmental and security requirements are imposed on deliveries using autonomous vehicles and drones, and this prevents their deployment at a larger scale (O'Flaherty et al., 2022). The risk consumers perceive challenges the societal acceptance of these two transportation means. According to a survey (Teale, 2022) conducted by Morning Consult, 66% of the respondents worried about air traffic safety due to drone usage. Moreover, 71% of the respondents answered they were concerned about personal or data privacy issues related to drone deliveries.

There have been several studies on the use of drones for SDD, as discussed in Section 2.5.2. However, Davis (2021) states that transportation and logistic companies are not yet ready to use autonomous vehicles and drones to deliver packages to customers' homes. Moreover, their impacts and challenges are currently unclear as there is no mass deployment yet, and the research on the safety impact is just emerging (Tafidis et al., 2022). Safety and security requirements should be integrated into the models to alleviate these concerns. As emphasized by Poikonen & Campbell (2021) and Bakach et al. (2022), future models should incorporate better representations of constraints and new features related to the use of drones or autonomous vehicles respectively. Moreover, behavioral economics models based on, e.g., willingness to accept, may be used to address societal concerns.

Sustainability: Faster may mean better for customers but not for the planet. Muñoz-Villamizar et al. (2021) show that fast deliveries increased total CO_2 emissions by up to 15% as they reduced the possibilities of cargo consolidation. However, the demand for faster shipping is growing, and the pressure on companies is high to avoid losing their hard-earned sales. Most scientific studies ignore this issue and focus solely on revenue-related aspects. To have a sustainable future, new delivery strategies that incorporate environmental aspects need to be envisaged. The environmental aspect should also be considered in model objectives to undertake environmental responsibility. To this aim, the first step is measuring the environmental impact of vehicles' operations in terms of emissions (Lin et al., 2018). Stakeholders should invest more in improving the efficiency of the delivery network, utilizing greener transportation means (like electric vehicles, bicycles, and porters), and/or setting up pickup stations. Also, getting customers aware of the topic contributes to sustainable deliveries. A possibility is to provide information while customers make choices to stimulate their eco-consciousness (Gleim et al., 2013). For example, Agatz et al. (2021) find that green labels that specify environmentally friendly time slots help motivate customers to choose greener time slots that reduce routing distance.

Fairness and road safety: One of the critical challenges in SDD is balancing limited resources while meeting customer expectations. Prioritizing high-profit orders may increase short-term profitability, but may risk overlooking fairness in service availability. By excluding low-profit orders, companies may erode customer loyalty, as those

customers might seek services from competitors. Some studies, such as Chen et al. (2023), propose maximizing the minimum service rate over all regions. Managers can adopt strategies from these studies by dividing customers into groups based on their willingness to pay and then maximizing revenue while ensuring a certain service level across all groups.

Besides, most papers consider profit-maximizing objectives and completely ignore the security of drivers and road users. According to a recent study (Center, 2022), drivers delivering parcels for Amazon Delivery Service Partners[1] were injured at nearly one injury per five full-time-equivalent workers in 2021. Gig workers often spend quite a long time on the road, whether rainy or sunny, and they try to complete as many orders as they can to get the incentives offered by companies. Several risk factors are causing safety issues, such as fatigue, rush to fulfill orders, and harsh weather conditions (Lim & Ong, 2022). These road safety issues affect every delivery driver regardless of the kind of transport used. To lower the risk, companies can implement policies and assist. An interesting alternative could be to develop algorithms leading drivers to follow a safer path or making them less stressful while delivering orders on time, for example, by taking into account the deadline of orders and other conditions like accident-prone areas, weather, riders' riding distance, and experience (to estimate fatigue level, Fu & Ma (2022)). Yildiz & Savelsbergh (2019a) propose a model that guarantees a minimum payment for drivers without significantly affecting the service level and other typical performance measures. Balanced and fair workload distribution among drivers is another topic that deserves more attention when developing future SDD services.

Focusing on methodology and optimization approaches for SDD services, we envisage the following future research directions:

Multiple objectives: In order to select the most proper model for the application under consideration, the impacts of decisions should be considered from different perspectives, such as customers' fairness, drivers' safety,

[1] The workers in Amazon's delivery system operate as direct employees at one of its facilities, send parcels as gig workers via Amazon's app Flex, or drive for one of Amazon's delivery contractors, called Delivery Service Partners.

quality of service, number of customers served, or sustainability. Thus, a multi-objective modeling approach would help decision-makers to consider different and sometimes conflicting goals simultaneously. This is particularly important for developing greener and sustainable models and the respective delivery strategies. While some of the recent studies employ tools of multi-objective optimization (Pureza & Laporte, 2008; Yildiz & Savelsbergh, 2019a; Chen et al., 2023; Wu et al., 2023), the full potential of this methodology for SDD services is still to be exploited.

Modeling uncertainty with two-stage or multi-stage stochastic programming models: As stated above, there are many uncertainty sources in SDD services. Many papers dealing with stochastic and dynamic problems propose MDP-based approaches fit for online decisions. However, for offline decisions or when combined with strategic and tactical decisions, the approaches using two-stage or multi-stage stochastic programming (Birge & Louveaux, 2011) are another alternative. The computational performance of two-stage models has recently been studied in Li et al. (2024b) and Ausseil et al. (2022), and the results show the great potential of these approaches.

Modeling SDD services with multiple decision-makers: When problems involve multiple decision-makers, one could consider them from the game theory perspective. Various SDD problems could be modeled as simultaneous or sequential games with collaborative or competitive players. For example, in the problem discussed in Section 2.4.1, which is to determine prices of delivery time slots to maximize the expected revenue, the provider first decides the price of each slot option for each individual, and then each customer selects an option based on his or her WTP. This can be translated into a sequential two-stage (Stackelberg) game: The first decision-maker (who acts as a leader) decides the prices, and then the multiple followers (customers) decide which time slot to accept (see, e.g., Cerulli et al. (2024)).

Empirical studies comparing pros and cons of different strategies: In Section 2.8, we compared different SDD problems (strategies) in terms of customer fairness, quality of service, and environmental aspects. While this comparison is conducted on small illustrative instances, more critical and comparative studies are indeed necessary in a much broader context of SDD services, to better understand the challenges and opportunities of different

strategies and provide a more solid base for stakeholders to choose their preferred strategy. Establishing a standard and publicly available benchmark set of instances is important for such empirical studies. In Table 2.6, we list available test data provided in the papers surveyed. The papers on the first three entries use data adapted from Solomon (1987); the last three on meal delivery are crafted based on real-world data. It is evident that there persists a lack of publicly accessible data sources.

Table 2.6: Available test data of papers surveyed.

Literature	Data url
Voccia et al. (2019)/Côté et al. (2023)	http://ir.uiowa.edu/tippie_pubs/65/
Li et al. (2024b)/Archetti et al. (2020a)	https://drive.google.com/file/d/1mVw2u1vuvq6NooKbfHlcYPQReBJXqimQ/view
Pina-Pardo et al. (2024a)	https://github.com/jcpinap/DTSP-DR
Zhang et al. (2023)	https://github.com/amflorio/dvrp-stochastic-requests
Ausseil et al. (2022)	github.com/bstabler/TransportationNetworks/tree/master/chicagoregional
Yildiz & Savelsbergh (2019a)/Auad et al. (2023)	https://github.com/grubhub/mdrplib
Mao et al. (2019)	https://doi.org/10.1287/msom.2022.1112

2.10 Conclusion

To compete with brick-and-mortar retailers, many e-retailers offer SDD services. Fast delivery services mainly aim to compensate for the lack of "instant gratification," which affects e-shoppers versus standard shoppers. These fast delivery services come with new challenges for distribution and logistics companies. This chapter focuses on the optimization problems in SDD services and analyzes the corresponding literature. Particularly, we show how researchers address the dynamic arrival times of orders (Section 2.3), decisions about time slot allocation for deliveries (Section 2.4), delivery options selection (Section 2.5), emerging problems in pickup and delivery systems (Section 2.6) and tactical districting decisions (Section 2.7). Moreover, we highlight the more representative problems emerging in SDD by providing formulations for deterministic versions. We also provide a comparison of solution structures obtained by applying the different formulations and discuss the corresponding implications. Finally, we present challenges and managerial insights in dealing with increased uncertainties, guaranteeing reliable deployment of innovative delivery means, environmental concerns, and road safety. We discuss potential research

directions associated with these challenges.

The objectives explored in the literature encompass critical aspects of optimizing service quality and financial performance, and the changes in the objectives or constraints may greatly impact the decisions. The objectives of a company may evolve across different stages of its lifecycle: at the startup stage, the key objective may be maximizing survival probability with limited resources; at the growth and maturity stages, the primary objectives may shift to increasing profitability or maximizing some performance measures. Moreover, for large companies, environmental and social responsibility should also be more important. Similarly, the constraints also vary across stages, resulting in different optimization problems to solve.

We conclude by noting that the research field associated with SDD services is evolving very fast, as it happens to the corresponding practical counterpart. Thus, developing interesting research results with a long-term impact in the field is hard, as new results might soon become obsolete. Still, the field offers a lot of challenges and opportunities to study and develop novel optimization tools for highly complex systems.

3

Reinforcement Learning Approaches for the Orienteering Problem with Stochastic and Dynamic Release Dates

joint work with **Claudia Archetti** and **Ivana Ljubić**

3.1 Introduction

E-commerce markets are booming at remarkable rates. Due to an unprecedented series of lockdowns, billions of people stayed at home to prevent the spread of COVID-19. It is reported that the e-commerce revenue saw a 10% increase in Europe in 2020 due to the pandemic (Statista, 2021). At the same time, the expectations of customers in terms of service quality are also increasing. When it comes to the specific challenges in improving customers' satisfaction with delivery, one of the crucial features is related to the last-mile delivery leg (Archetti & Bertazzi, 2021). An important feature of last-mile distribution is that its operations start as soon as parcels are delivered to the final logistic center, typically a city distribution center (CDC). Given the short delivery times requested by the customers, delivery operations typically need to start before all parcels expected to arrive during the day are available at CDC. This raises a question: should the dispatcher wait for more or all parcels to be delivered at CDC, or should he/she start the delivery as soon as there is any available parcel and vehicle?

In this chapter, we focus on the sequential decision-making problem related to when to deliver parcels and which parcels to deliver under the assumption that the time at which parcels become available at CDC (called *release dates*) is stochastic and dynamic. Release dates are the moments when parcels become available for delivery. On the other hand, the release time is like the beginning of a time window, i.e., the earliest feasible time at which a customer can be visited or a cargo can be picked up. In problems with release time, a vehicle can start the route without waiting for all parcels to be ready. We introduce a new problem called the *Orienteering Problem with Stochastic and Dynamic Release Dates* (DOP-rd). The problem finds applications in same-day delivery (SDD) systems (Li et al., 2024a). As stated in Stroh et al. (2022), SDD services face tight delivery deadlines and relatively low order volumes, thus time instead of vehicle capacity tends to be the limiting resource. We assume that a single and uncapacitated vehicle is serving customer requests with no time window so that the vehicle should finish its service before the deadline T_E, which might correspond to the duration of the driver's working shift. The release dates are considered to be uncertain and their distributions are dynamically updated during the sequential decision-making process. The

objective is to maximize the expected number of requests served within T_E. This corresponds to maximizing the expected number of customers who get their packages within the desired deadline (end of the day), thus in turn, this implies the maximization of customer satisfaction (Joerss et al., 2016; Voccia et al., 2019).

This chapter presents the following main contributions to the literature:

1. We introduce and study DOP-rd. The problem shares similarities (with respect to the input setting) with the one studied in Archetti et al. (2020a). The main differences are in the objective function and the presence of a deadline. While Archetti et al. (2020a) aim to minimize the expected traveling and waiting time while ensuring that all parcels are delivered, the DOP-rd seeks to maximize the expected number of parcels delivered within a deadline T_E. The shift in objectives and constraints brings fundamental changes to the problem setting, necessitating unique solution approaches.

2. We model the problem as a Markov Decision Process (MDP) and propose a batch approach to approximate future routes (actions), based on Monte-Carlo simulation for scenario generation and continuous approximation from Daganzo (1984) for estimating the duration of routes. To the best of our knowledge, the method we propose to approximate future routes has never been applied before to routing problems. At each decision epoch, we learn the value function based on the updated information, which is embedded both in scenario generation and in the approximation approach. Thus it leads to a reinforcement learning approach.

3. We propose two novel hybrid approaches to derive a decision-making policy in the (approximated) decision space. Both belong to the class of value function approximation (VFA) methods and are hybridized with a one-step look-ahead policy. The first is combined with a consensus function (VFA-CF) and the second with a two-stage stochastic ILP model (VFA-2S). The former approach relies on exact solutions of a series of deterministic integer linear programs (ILPs); the latter one uses a two-stage stochastic ILP model instead. These ILPs are solved using state-of-the-art branch-and-cut techniques. The obtained solutions are then

used to decide when to leave the depot and which customers to serve while maximizing the expected number of parcels delivered. In addition, we propose a partial characterization of optimal policy, the conditions of which are checked before applying VFA-CF/VFA-2S.

4. Most of the existing literature on stochastic and dynamic vehicle routing problems dealing with MDP relies on heuristic policies (see Ulmer et al. (2020)). In this work, we approximate future routes in each decision epoch and employ exact methods to determine the best action associated with the current state. Thanks to this advantage of embedding exact methods inside a reinforcement learning framework, VFA-CF/VFA-2S can be run online without sophisticated learning procedures that require extensive training, such as those based on neural networks. Specifically, learning the value function requires only minimal tuning of a few hyper-parameters.

5. In an extensive computational study, we show VFA-CF/VFA-2S excel other benchmark approaches thanks to these following distinct features: (1) embedding of the future information, (2) using sampling rather than point estimations, (3) using exact methods rather than simple heuristics, and (4) using approximations of future routes. Specifically, concerning point (2), we compare the VFA approaches with a fully deterministic approach, which builds exact solutions based on point estimations of future release dates. We additionally provide results of a competitive analysis, in which the upper bounds are calculated by solving to optimality the problem under the assumption of having perfect information. Besides, we conduct experiments with two vehicles to test the potential and adaptability of proposed VFA approaches when extended to the multi-vehicle case.

The rest of the chapter is organized as follows. Section 3.2 presents a literature review by focusing on the differences with the current work. Section 3.3 contains the problem description and formulation as a Markov Decision Process. Section 3.4 presents upper bounds to solution values and a partial characterization of the optimal policy. The solution approaches are introduced in Section 3.5, while Section 3.6 is devoted to computational experiments

and results. Finally, we conclude in Section 3.7.

3.2 Literature review

We first focus on the literature on routing problems with stochastic and dynamic features and orienteering problems in particular, and then on the common approaches of reinforcement learning used for these routing problems. Finally, we move to problems with release dates and highlight differences in problem settings and solution approaches compared to the current work.

The literature on stochastic and dynamic routing problems is wide. We focus on pioneering papers and surveys. Psaraftis (1988) introduce the traveling salesman problem with the dynamic customer demands. Bertsimas & Van Ryzin (1991) generalize the former results and propose a mathematical model for the stochastic and dynamic vehicle routing problems (VRPs). Bent & Van Hentenryck (2004) propose using the multi-scenario sampling in combination with a consensus function for the dynamic vehicle routing problem with time windows. Other interesting extensions of stochastic or dynamic settings appear in Savelsbergh & Sol (1998), Secomandi (2001), Secomandi & Margot (2009), Goodson et al. (2013), Subramanyam et al. (2021). As for surveys on stochastic and dynamic VRPs, we refer to Ritzinger et al. (2016), Oyola et al. (2018), Soeffker et al. (2022).

The problem we study belongs to the class of orienteering problems. An Orienteering Problem (OP) involves a set of nodes, each with a corresponding score. The goal is to determine a path (or a tour) that visits a subset of these nodes, maximizing the total accumulated score subject to a limited time budget. For the details on the variants and applications, we refer readers to surveys by Vansteenwegen et al. (2011), Archetti et al. (2014), and Gunawan et al. (2016). Specifically, Vansteenwegen et al. (2011) present extensions, variants, solution approaches, and applications. Archetti et al. (2014) present a survey of the class of vehicle routing problems with profits, where the orienteering problems are considered as the basic problems. Gunawan et al. (2016) extend the summaries of

Vansteenwegen et al. (2011) and Archetti et al. (2014) by including more recent papers and new variants. Gunawan et al. (2016) also study OP with stochastic aspects such as uncertainties in the collected scores, travel and service times, waiting time, etc. In addition, Zhang et al. (2018) present a variant of OP in which the traveler may experience stochastic waiting time at locations. The objective is to maximize the expected rewards collected by determining the visiting sequence while respecting each customer's time window. Song et al. (2020) address a problem involving regular service for subscription customers. The goal is to minimize the expected costs minus the revenue from serving on-demand customers. The problem is modeled as a two-stage stochastic decision problem, where the first stage assigns drivers to customers and the second stage is a team-orienteering problem with time windows. Another paper by Angelelli et al. (2021) studies a dynamic and stochastic variant of OP spanning two continuous days. On the first day, the company accepts or rejects real-time requests that arrive randomly. On the second day, the accepted requests require a vehicle to visit customers for goods pickup. The goal is to maximize the expected profit, defined as the difference between the expected total prize and travel cost.

3.2.1 Reinforcement Learning for Stochastic and Dynamic Routing Problems

Reinforcement learning (RL) has emerged as a successful approach for solving various decision-making problems (Silver et al., 2016). It is based on the fundamental concept that an agent interacts with an environment and learns what to do based on the feedback it receives in the form of rewards or penalties. The detailed techniques and algorithms used in RL have been extensively discussed in books, with notable references including Sutton & Barto (2018) who describe the topic from the point of view of computer science, Bertsekas (2019) who adopts the language of control theory, and Powell (2022) who names it approximate dynamic programming (ADP) using the language of operations research. The application of RL in solving stochastic and dynamic routing problems has gained significant attention. Next, we present the related research works.

Q-learning is an algorithm that learns the value of taking an action in a state. It is one of the most popularly used

RL algorithms and has been applied to address various routing problems. Here are some examples. Mao & Shen (2018) study the adaptive routing problem in stochastic and time-dependent traffic networks. The goal is to find a routing strategy to minimize the expected total travel time. To solve the problem, they introduce two variants of Q-learning approaches. The first variant is an online Q-learning method that requires a discrete state space. The second is an offline fitted Q iteration technique with a continuous state space that approximates the Q function to overcome the curse of dimensionality. Basso et al. (2022) consider a VRP with a single electric commercial vehicle for reliable charge planning. The problem considers the uncertainties in energy consumption and customer requests, aiming to minimize the expected energy consumption. To address the problem, the authors propose a combination of Q-learning and VFA techniques. Specifically, they employ Q-learning with expected energy and battery depletion and a VFA based on a reduced state representation. Revadekar et al. (2020) study a pickup and delivery problem in the context of online medicine delivery. In the problem, customers order medicines through an online system, which finds the nearest pharmacy and assigns a delivery person to serve the delivery request. The goal is to minimize the delivery time. To tackle the problem, the authors propose a double Q-learning, which uses a double estimator to update the Q tables storing distance and time reward matrices separately. Similarly, Bozanta et al. (2022) study a pickup and delivery problem but in a meal delivery service. The problem involves picking up meals from customer-requested restaurants and delivering them to customers located on an $m \times m$ grid, with a limited number of available couriers and a restricted time frame. In this case, the objective is to maximize the revenue derived from the requests served. The authors propose three different Q-learning approaches. Chen et al. (2023) work on addressing the unfairness issue in SDD service availability. They model the problem with a weighted sum of two objectives: maximizing the overall service level (utility) and maximizing the minimum service rate across all regions (fairness). To address this problem, the authors first use a routing heuristic to reduce the action space and then propose a deep Q-learning approach. Chen et al. (2022) also utilize a deep Q-learning approach but in the context of a decision problem involving the assignment of new customers to either drones or vehicles or the rejection of service. For more references on the topic, we refer readers to the survey by Hildebrandt et al. (2023).

Contrary to the methods described above, we do not rely on classical Q-learning mechanisms, and employ exact methods combined with a one-step look-ahead and approximation of future routes instead. To the best of our knowledge, the closest to our work in the routing literature is the study by Anuar et al. (2021) where the authors study a multi-depot dynamic VRP with stochastic road capacity caused by earthquake damages. In their problem setting, the routes computed in advance may become infeasible during real-time execution due to unforeseen disasters such as earthquakes. The authors propose a solution approach using two ILP models, one for replenishment and the other for delivering supplies. These two models are used interchangeably based on the vehicle's capacity. Like the current work, Anuar et al. (2021) use exact methods inside the RL/ADP framework. However, the two works differ in several key aspects: 1) Problem domain and uncertainty factors: Anuar et al. (2021) work on humanitarian applications, whereas our paper focuses on same-day deliveries. Furthermore, in the approach proposed by Anuar et al. (2021), the computation of a complete route at each decision epoch aims at incorporating the uncertain road capacity. In contrast, we aim at maximizing the expected number of requests while incorporating the uncertainty of release dates, a prevalent aspect in SDD business practices; 2) Model complexities: The models proposed by Anuar et al. (2021) primarily focus on determining the next destination for each vehicle at each decision epoch. In contrast, our models calculate multiple future routes, and the decision made pertains to building a complete route. This difference in the modeling approach leads to variations in the complexity and structure of the ILP models employed; 3) RL approaches: Anuar et al. (2021) use a rollout approach, whereas we employ VFAs with a one-step look-ahead. Notably, the idea of approximating future routes is not employed by Anuar et al. (2021).

3.2.2 Routing Problems with Release Dates

The previous studies mainly focus on demands and travel time uncertainties. The works on other stochastic and dynamic aspects, such as release dates, appeared recently. The term "routing problem with release date" was first introduced in Cattaruzza et al. (2016), where the authors define it as a problem where each parcel is associated with

a release date indicating the time at which the parcel is available at depot. Release dates are supposed to be known beforehand. The problem is a multi-route vehicle routing problem with release dates, and the authors solve it through a population-based algorithm. In Archetti et al. (2015), the Traveling Salesman Problem with release dates (TSP-rd) is considered, i.e., the problem with a single uncapacitated vehicle. The authors analyze the complexity of the problem in two variants - minimizing the total distance traveled with a constraint on the maximum deadline and minimizing the total time needed to complete the distribution without a deadline. Later, Reyes et al. (2018b) extend the results found in Archetti et al. (2015) by considering service guarantees on customer-specific delivery deadlines. Shelbourne et al. (2017) consider VRP with release and due dates where a due date indicates the latest time the order should be delivered to the customer. The objective function is defined as a convex combination of the operational costs and customer service level. The authors propose a path-relinking algorithm to address the problem. More recently, Archetti et al. (2018) propose a mathematical programming formulation and present a heuristic approach based on an iterated local search for solving TSP-rd.

All contributions mentioned above study routing problems with deterministic release dates. Recently, there have been several studies related to applications in SDD services, where release dates are stochastic. Among them, Voccia et al. (2019) study a multi-vehicle dynamic pickup and delivery problem with time constraints where requests arrive dynamically from online customers. Each request is associated with a deadline or a time window. The objective is to maximize the expected number of requests that can be delivered on time. The authors first model the problem as an MDP, then, based on sample-scenario planning, they propose a heuristic solution approach. Rather than defining delivery windows at the receiver side, Van Heeswijk et al. (2019) consider dispatch windows at the consolidation center. As in Voccia et al. (2019), an MDP is defined. To overcome "the three curses of dimensionality", the authors propose an ADP algorithm using a VFA. In contrast, Archetti et al. (2020a) study a single-vehicle problem with no deadline and where the objective is to deliver all parcels as fast as possible. They propose a reoptimization approach that heuristically reduces the number of decision epochs. Furthermore, two models are defined to select an action at each reoptimization epoch. While both models proposed in Archetti et al.

(2020a) aim to minimize the total completion time, the main difference between the two is that the first model is a stochastic mixed-integer program model that considers the full stochastic information on the release dates, while the second one is a deterministic model that uses a point estimation (PE) for the stochastic release dates. For a detailed survey on routing problems in SDD, see Li et al. (2024a).

Based on Archetti et al. (2020a), our paper studies the DOP-rd with a single, uncapacitated vehicle serving customer requests with no time window. Unlike Archetti et al. (2020a), which does not consider drivers' working hours or strict deadlines, we aim at maximizing the expected number of parcels delivered within a deadline T_E. Recent studies highlight the practical importance of focusing on the maximum number of delivered parcels (Voccia et al., 2019; Behrend & Meisel, 2018; Ulmer & Thomas, 2018), as vehicles often need to return to the depot during working hours, and some parcels may be deferred for distribution the next day or by a third-party carrier. In Archetti et al. (2020a) every solution in which all parcels are served is feasible. In their setting, the vehicle can wait until the last parcel arrives at the depot and subsequently serve all the parcels in a single route. On the contrary, in our model, we address strict deadline constraints, ensuring timely parcel delivery. This distinguishes our approach from Archetti et al. (2020a).

As for solution approaches, the decision-making policy we consider adds to existing reoptimization frameworks behind the VFA approaches in routing literature. Indeed, instead of proposing a heuristic approach for solving problems in each scenario as in Voccia et al. (2019) or at each reoptimization epoch as in Archetti et al. (2020a), we solve ILPs with the approximation of future routes to balance the computational efficiency and policy performance. Our extensive numerical study shows the value of our proposed reoptimization scheme in routing applications. The ones proposed in the current work are innovative not only in the way future routes are approximated but also in how policies are derived through VFA-2S and VFA-CF and in the way the ILP models are constructed.

Another contribution, which is very close to the current study, is given by Van Heeswijk et al. (2019). Again, the problem is formulated as an MDP. However, decision epochs are predefined time instants separated by equidistant

intervals, as in Klapp et al. (2018b). The objective function in Van Heeswijk et al. (2019) is to minimize the expected total dispatching costs. The paper focuses on dispatching decisions, so the routing decisions are not considered in the optimization problem, and routing costs are estimated by Daganzo's formula. In our work, decision epochs are set dynamically according to the release dates. Also, differently from Van Heeswijk et al. (2019), we consider a single vehicle. Finally, we use Daganzo's formula with a different scope, i.e., for inferring how many requests can be fulfilled in future routes at each decision epoch. Instead, we get the exact routing sequence for the current epoch.

Finally, another related work is given by Schrotenboer et al. (2021), where the authors study how to balance the consolidation potential and delivery urgency of orders. Contrary to our setting, it is a pickup and delivery problem with a fleet of vehicles and customer deadlines, aiming to maximize the expected customer satisfaction. Their approach is based on a cost function approximation.

We summarize the major characteristics of the contributions on routing problems with release dates in Table 3.1. We emphasize that we define release dates as *dynamic* if the information regarding their distribution is updated throughout the day. The papers listed mostly study dynamic problems where "dynamic" means customer orders are not fully known in advance and arrive dynamically at the CDC, but few of them consider updated release dates.

3.3 Problem description

The DOP-rd is defined as follows. A company distributes parcels to its customers from a CDC. At CDC the company may hold some level of inventory, so some parcels may be available right away, but most of them are delivered to the CDC by suppliers throughout the day. Hence, the distribution of the latter parcels can start only after their arrival at CDC. The moment at which parcels are delivered to the CDC is unknown. In case suppliers' vehicles are equipped with Global Positioning System (GPS), the information about their arrivals to the CDC

Table 3.1: Literature Comparison

		Release dates					
Literature	**Vehicle**	**Stochastic**	**Dynamic**	**Deadline**	**Objective**	**MDP**	**Approach**
Cattaruzza et al. (2016)	Single	✗	✗	✓	Min(total travel distance)	✗	Hybrid genetic algorithms
Archetti et al. (2015)	Single	✗	✗	✓	Min(total travel distance within deadline) and	✗	-
	Unlimited	✗	✗	✗	Min(the completion time)	✗	-
Reyes et al. (2018b)	Single	✗	✗	✓	Min(the completion time) and Min(total travel distance within deadline)	✗	-
	Unlimited	✗	✗	✓	Min(the travel distance)	✗	-
Shelbourne et al. (2017)	Multi	✗	✗	✓	Min(a convex combination of the total distance traveled and the total weighted tardiness of delivery)	✗	A path-relinking algorithm based on a hybrid evolutionary framework.
Archetti et al. (2018)	Single	✗	✗	✗	Min(the completion time)	✗	A heuristic approach based on an iterated local search
Archetti et al. (2020a)	Single	✓	✓	✗	Min(the expected completion time)	✓	Two models proposed (stochastic and deterministic), the solution is obtained through an iterated local search
Voccia et al. (2019)	Multi (plus a third party)	✓	✗	✓	Max(the expected #requests served)	✓	Sample-scenario planning and a variable neighborhood search implemented for each scenario
Van Heeswijk et al. (2019)	Multi (plus a third party)	✓	✗	✓	Min(the expected total dispatching costs)	✓	Approximate dynamic programming using a linear value function approximation to estimate the downstream costs
Klapp et al. (2018b)	Single	✓	✗	✓	Min(the expected total travel costs plus penalties for unattended and realized orders)	✓	Three heuristic policies(a priori, direct rollout, roll out a prize-collecting TSP guided by an initial a priori solution)
Schrotenboer et al. (2021)	Multi	✓	✗	✓	Max(the expected customer satisfaction)	✓	A cost function approximation approach that modifies a set-packing formulation
This chapter	Single and two vehicles	✓	✓	✓	Max(the expected #requests served)	✓	VFA-CF and VFA-2S approaches proposed relying on the approximation of future routes

(also called depot in the following) is constantly updated. In SDD, customer locations can be partitioned into districts with dedicated vehicles, where each district can be served by a vehicle multiple times during working hours (Banerjee et al. (2022), Stroh et al. (2022)). We consider a single uncapacitated vehicle to perform the service within a district. The goal is to serve as many parcels as possible within a given deadline.

More formally, let $G = (V, A)$ be a complete graph. The set of vertices V includes vertex 0, representing the depot, and the set N of customers. Each customer is associated with an arrival time of its parcel to the depot, called release date, and a delivery location. Packages arrive throughout the day until a predefined deadline T_E, corresponding to when the vehicle has to be back at the depot and finish the service. Release dates are stochastic, and their distributions are dynamically updated throughout the day. Specifically, the stochastic aspect simulates the situation where suppliers' vehicles carrying the parcels to the CDC might encounter unpredicted events. The dynamic aspect is related to the fact that the information about the arrival of the suppliers' vehicles to the CDC evolves over time, as in the case of GPS-equipped vehicles. Each arc $(i, j) \in A$ is associated with a traveling time $d_{ij} > 0$, and we assume that the triangle inequality holds, i.e., $d_{ij} + d_{ji'} \geq d_{ii'}, i, j, i' \in V$. A single uncapacitated vehicle is available to perform the deliveries. We define a route as a tour starting and ending at the depot and visiting customers in between. The set of routes is denoted as $\mathcal{K} = \{0, ..., |\mathcal{K}|-1\}$, where $\mathcal{K}$ contains the first route leaving for delivery (also called route 0), followed by the set of future routes denoted by K. The goal is to maximize the expected number of requests served within T_E.

In this chapter, we present the case in which the orders are placed, but the arrival time of the parcels ordered at the depot (CDC) is unknown. However, the solution approach we propose is also valid for the case where goods are at the depot but the request time is unknown, as studied in Voccia et al. (2019). In both cases, the customer locations are assumed to be known. We note that the problem studied in Voccia et al. (2019) represents a particular case of our problem as it does not consider dynamic release dates.

We now formalize the problem as a Markov Decision Process (MDP). The detailed notation used is summarized

in Table 3.2.

Table 3.2: Overview of notation used in problem introduction

	Problem Description
N	Set of customers
V	Set of vertices, $V = \{0\} \bigcup N$
A	Set of arcs connecting nodes i, j, where $i, j \in V$
G	A complete graph, $G = (V, A)$
T_E	Deadline, i.e., when the vehicle has to be back to the depot, and no further deliveries are allowed
d_{ij}	Traveling time from i to j, where $i, j \in V$
$\mathcal{K}$	Set of routes, $\mathcal{K} = \{0, ..., \lvert\mathcal{K}\rvert - 1\}$, including route 0 and the set of future routes denoted by K
e	Decision epoch, $e \in \{0, ..., E\}$
t_e	Time at decision epoch e
N_e^{served}	Set of customers already served before decision epoch e
$N_e^{unserved}$	Set of customers unserved at decision epoch e
N_e^{known}	Set of unserved customers whose parcels are available at the depot at decision epoch e
$N_e^{unknown}$	Set of unserved customers whose parcels are still to be delivered to the depot
N_e^{static}	Set of customers whose information about release date is not updated until their parcels arrive at the depot
$N_e^{dynamic}$	Set of customers whose information about release date is dynamically updated
$\widetilde{r}_i^e$	Random variable associated with the release date of customer $i \in N_e^{unserved}$ at decision epoch e
r_i	Actual arrival time of the parcel of customer $i \in N_e^{known}$
	Components of MDP
φ	Time interval between two additional decision epochs
S_e	State at decision epoch e, $S_e = (t_e, N_e^{known}, \{\widetilde{r}_i^e\}_{i \in N_e^{unknown}})$
$X(S_e)$	Action space at decision epoch e
$\mathcal{X}_e$	Decision/action at decision epoch e, $\mathcal{X}_e \in X(S_e)$
x_{ij}^e	Binary arc variable at epoch e, taking value 1 if the route traverses arc (i, j), 0 otherwise, where $i, j \in N_e^{known} \cup \{0\}$
$l(\mathcal{X}_e)$	Set of customer locations visited in the route associated with the action $\mathcal{X}_e$
$t_{route}(\mathcal{X}_e)$	Time required to perform the route associated with action $\mathcal{X}_e$
t_p	Earliest time when a new parcel arrives while the vehicle is at the depot
N_{e+1}^{new}	Customers whose parcels arrived at the depot between decision epochs e and $e+1$
$V_e(S_e)$	The value at decision epoch e with state S_e
$C(S_e, \mathcal{X}_e)$	Number of requests served by action $\mathcal{X}_e$
γ	Discount factor
	Upper Bounds
$\lvert\mathcal{K}\rvert$	Number of routes performed by the vehicle
s_i^k	Binary variable for route $k \in \mathcal{K}$, equal to 1, if customer or depot $i \in V$ visited, 0 otherwise
ξ_{ij}^k	Binary arc variable for route $k \in \mathcal{K}$, equal to 1, if customer $i \in N$ visited, 0 otherwise
d_k	Starting time of route $k \in \mathcal{K}$
α_i	Binary variable equal to 1, if a direct route from the depot to customer $i \in N$ and back, is performed, 0 otherwise

3.3.1 The Problem as a Markov Decision Process

At each decision epoch e, we distinguish between customers that have already been served (i.e., their parcels have been delivered), N_e^{served}, and those that are still unserved, $N_e^{unserved}$, where $N_e^{served} \bigcup N_e^{unserved} = N$. There are two categories of unserved customers based on the information available for the release date at decision epoch e. If the parcel has arrived at the depot, the release date is known, and the customer can be served, thus we call this set of customers N_e^{known}. On the contrary, $N_e^{unknown}$ denotes the set of customers whose parcels are still to be delivered to the depot. In turn, there are two subsets of $N_e^{unknown}$: customers whose information about the distribution of the release date is not updated until the parcel arrives at the depot are denoted as N_e^{static}; customers whose information about the distribution of the release date is updated, i.e., a new estimation of release date distribution is given in real-time, thanks to GPS equipped vehicles, are denoted as $N_e^{dynamic}$. Note that $N_e^{static} \bigcup N_e^{dynamic} = N_e^{unknown}$. The parcels are delivered individually or in batches from suppliers to the depot. Let $\widetilde{r}_i^e$ be the random variable associated with the release date of customer $i \in N_e^{unserved}$ at decision epoch e. This variable is defined as follows:

- for $i \in N_e^{known}$, $\widetilde{r}_i^e = r_i$, where r_i is the actual arrival time of the parcel of customer i.
- for $i \in N_e^{static}$, we have $\widetilde{r}_i^e = \widetilde{r}_i^0$, which means that the distribution of the random variable associated with the release date is estimated at time 0 and never updated till the parcel arrives at the depot. In practice, the mean and variance of the distribution can be empirically obtained from historical data.
- for $i \in N_e^{dynamic}$, $\widetilde{r}_i^e$ is a random variable representing release date at decision epoch e. Its distribution is dynamically updated throughout the day.

The distinction between $N_e^{dynamic}$ and N_e^{static} does not relate to policy design (which is solely based on $N_e^{unknown}$) but rather pertains to the utilization of instances generated for numerical tests (see Section 3.6.1).

At each decision epoch e, the company has to decide whether to wait for more parcels to arrive or to dispatch

the vehicle delivering a subset of the parcels available at the depot. No preemptive return is allowed, i.e., the vehicle comes back to the depot only once all parcels on board are delivered to the corresponding customers. Ulmer et al. (2019b) investigate the benefit of allowing preemptive depot returns to integrate dynamic requests into delivery routes. The results showed that benefits are marginal.

The MDP components are the following:

- *Decision epochs:* A decision epoch is denoted as $e \in \{0, \ldots, E\}$, and the time associated with it is t_e. Note that T_E represents the deadline, i.e., when the vehicle has to be back to the depot, and no further deliveries are allowed. The first decision epoch 0 corresponds to the beginning of the delivery operations. At each decision epoch, an action is made based on the available deterministic and updated stochastic information. A decision epoch corresponds to either the arrival of a package at the depot or to the time when the vehicle is back to the depot at the end of a route. To avoid waiting for too long at the depot without having any new arrival of packages, additional decision epochs are defined at equally distant time intervals φ, when the action is evaluated according to the updated stochastic information.

- *States:* The state $S_e = (t_e, N_e^{known}, \{\widetilde{r}_i^e\}_{i \in N_e^{unknown}})$ includes all information for evaluating an action at decision epoch e, i.e., the time of the decision epoch t_e, the set N_e^{known} of customers with known release dates (whose parcels have arrived), and the stochastic information (i.e., distributions) related to random variables associated with future release dates $\{\widetilde{r}_i^e\}_{i \in N_e^{unknown}}$. At decision epoch 0, no customer has been served.

- *Actions:* At each decision epoch e, the company has to either dispatch a subset of accumulated orders or wait at the depot for new parcels to arrive. The decision involves action $\mathcal{X}_e$, to which we also refer as route 0 for decision epoch e. When the vehicle is not at the depot or if the vehicle is at the depot, but there is no parcel available, the decision is to wait, i.e., the route 0 is empty. Similarly to what has been done in Archetti et al. (2020a), if dispatching is decided, an action consists of a new route serving a subset of the unserved customers. However, while in Archetti et al. (2020a) customers served in the new route could belong to

either N_e^{known} or $N_e^{unknown}$, we instead reduce the set of customers to be potentially included in the new route to N_e^{known}. This implies that the new route, in case at least one customer is included, will leave immediately from the depot, contrary to Archetti et al. (2020a) where the vehicle might have to wait at the depot in case the route includes at least one customer in $N_e^{unknown}$. Formally, the action space at decision epoch e, denoted by $X(S_e)$, includes all possible subsets of N_e^{known}, each one associated with the TSP route serving all parcels in the subset. The set of customer locations visited in the route associated with the action $\mathcal{X}_e$ is denoted as $l(\mathcal{X}_e)$, which is empty if the route is empty, i.e., the vehicle is not dispatched and the action is waiting at the depot. The route o associated with the action in decision epoch e consists of a set of consecutive arcs traversed by the vehicle (this set could also be empty). The traversal of an arc is represented by the binary variable denoted as $x_{ij}^e, i, j \in N_e^{known} \cup \{0\}$. Thus $\mathcal{X}_e$ can be expressed in terms of these decision variables as $\mathcal{X}_e = (x_{ij}^e)$.

- *Transitions:* After action $\mathcal{X}_e$, a transition is made from state S_e to a new state S_{e+1}. The state S_{e+1} is described as $S_{e+1} = (t_{e+1}, N_{e+1}^{known}, \{\tilde{r}_i^{e+1}\}_{i \in N_{e+1}^{unknown}})$. We have $t_{e+1} = t_e + t_{route}(\mathcal{X}_e)$ if route is not empty, $t_{e+1} = \min(t_p, t_e + \varphi)$ otherwise, where $t_{route}(\mathcal{X}_e)$ is the time associated with the route corresponding to action $\mathcal{X}_e$, t_p is the earliest time when a new parcel arrives while the vehicle is at the depot, and φ is the maximum waiting time between two decision epochs as defined earlier. Let N_{e+1}^{new} be customers whose parcels arrive at the depot between decision epochs e and $e+1$. Correspondingly, the set of customers is updated as $N_{e+1}^{unknown} := N_e^{unknown} \backslash N_{e+1}^{new}$ and $N_{e+1}^{known} := N_e^{known} \backslash l(\mathcal{X}_e) \cup N_{e+1}^{new}$.

- *Objective:* The objective is to find a policy maximizing the expected number of requests served within T_E. Each action $\mathcal{X}_e$ chosen when the system is in state S_e, is guided by a decision rule and brings an *immediate reward*, expressing the number of requests served by the route dispatched at time t_e, and a future reward. A sequence of decision rules is defined as a *policy*, and the optimal policy is the one that maximizes the expected number of parcels delivered till epoch E.

  In decision epoch e, the goal is to maximize the expected number of requests served within the time interval

$[t_e, T_E]$. We set the final value function $V_E(S_E) = 0$, and for $e = 0, \ldots, E-1$, we define the value function as the following recursive formula:

$$V_e(S_e) = \max_{\mathcal{X}_e \in X(S_e)} \{C(S_e, \mathcal{X}_e) + \gamma \mathbb{E}[V_{e+1}(S_{e+1} \mid S_e, \mathcal{X}_e)]\}. \tag{3.1}$$

Here, $C(S_e, \mathcal{X}_e)$ denotes the immediate reward, and $\gamma \mathbb{E}[V_{e+1}(S_{e+1} \mid S_e, \mathcal{X}_e)]$ is the expected value associated with decision epoch $e+1$, while γ is a discount factor. In situations characterized by high information uncertainty and an inability to guarantee the precision of future reward estimations, a discount factor can be applied to discount the effect of making mistakes in the future (Burk & Averbeck, 2023). Considering the nature of the SDD problem, where time constraints are so tight, the more distant the future is, the less certain and accurate our solutions tend to be. Therefore, including a discount factor, as an algorithmic parameter, allows us to explicitly adapt to and manage the diminishing accuracy over time, providing better flexibility and performance of the algorithm in handling such a dynamic problem.

The system suffers from the curse of dimensionality in the number of states and actions, which grows exponentially with the number of requests. In the recent literature, reinforcement learning is frequently used to deal with this issue (see, e.g., Powell (2009) and Van Heeswijk et al. (2019)). In this work, we propose two reinforcement learning approaches for solving the DOP-rd, the first based on a value function approximation with a consensus function and the second on a value function approximation with a two-stage stochastic ILP model. Both methods are hybridized with a one-step look-ahead strategy, in which future release dates are sampled in a Monte-Carlo fashion.

The two approaches, which make use of branch-and-cut-based exact methods to improve the quality of decisions, are described in detail in Section 3.5.

3.4 Upper bounds and partial characterization of optimal policy

In the following, we discuss valid upper bounds useful for competitive analysis. We then provide sufficient conditions under which it is optimal for the vehicle to leave the depot immediately in order to serve a subset of available parcels.

3.4.1 Upper Bounds

A simple upper bound for the DOP-rd can be obtained by solving a special instance of the Orienteering Problem (OP). OP is defined on a complete graph $\hat{G} = (\hat{V}, \hat{A})$ in which we are given arc weights $\hat{w}_{ij} > 0$, $(i,j) \in \hat{A}$, and node prizes $\hat{p}_i > 0$, $i \in \hat{V} \setminus \{0\}$, where 0 is the depot. The goal is to find a route whose total arc weight does not exceed a budget $\hat{B} > 0$, and the total collected prize from visited nodes is maximized.

In each decision epoch e, we can calculate a valid upper bound on the maximum number of parcels that can be delivered within the interval $[t_e, T_E]$ in the DOP-rd. This bound is given as a solution of the OP over the set of parcels in $N_e^{unserved}$, with all parcel prizes being set to one, and the maximum budget set to $T_E - t_e$.

Such obtained upper bound can be used in a competitive analysis. Moreover, it gives rise to the partial characterization of the optimal policy described in Section 3.4.2. The methodology used for solving the OP is based on a branch-and-cut (see, e.g., Fischetti et al. (1998)), which is one of the most effective exact approaches for the OP.

In the following, we instead describe a stronger upper bound obtained a-posteriori when all information is known.

Upper Bound with Perfect Information: A tighter upper bound for DOP-rd can be computed in a *wait-and-see* fashion, using the perfect information with respect to realized release dates. In this case, the optimal solution is the solution of the OP with Release Dates (OP-rd), a deterministic counterpart of the problem studied in this article. Inspired by the model in Archetti et al. (2018) for TSP-rd and completion time minimization, we propose a formulation for OP-rd as follows.

Let us assume that all release dates, $r_i \geq 0, i \in N$ are known. Then, OP-rd aims to find the maximum number of parcels that can be served in the interval $[0, T_E]$. A solution consists of up to $|\mathcal{K}|$ routes performed by the vehicle, so that the end time of the last route does not exceed T_E. The problem is NP-hard since the OP can be reduced in polynomial time to it. We can model the problem as an ILP, making use of the following decision variables:

- s_i^k: binary variable indicating whether parcel i is served in route k;
- ξ_{ij}^k: binary variable equal to 1 in case arc (i,j) is traversed in route k;
- d_k: continuous variable representing the starting time of route k.

The ILP formulation for the OP-rd is then:

$$\max \sum_{i \in N} \sum_{k \in \mathcal{K}} s_i^k \tag{3.2a}$$

$$\text{subject to:} \quad \sum_{(i,j)\in A} \xi_{ij}^k = \sum_{(j,i)\in A} \xi_{ji}^k = s_i^k \qquad i \in V, k \in \mathcal{K} \tag{3.2b}$$

$$\sum_{i,j\in S} \xi_{ij}^k \leq \sum_{i\in S\setminus \ell} s_i^k \qquad S \subseteq N, |S| \geq 2, \ell \in S, k \in \mathcal{K} \tag{3.2c}$$

$$\sum_{k\in\mathcal{K}} s_i^k \leq 1 \qquad i \in N \tag{3.2d}$$

$$d_k \geq r_i s_i^k \qquad i \in N, k \in \mathcal{K} \tag{3.2e}$$

$$d_{k+1} = d_k + \sum_{(i,j)\in A} d_{ij} \xi_{ij}^k \qquad k \in \mathcal{K} \tag{3.2f}$$

$$d_{|\mathcal{K}|} = T_E \tag{3.2g}$$

$$\xi_{ij}^k, s_i^k \in \{0,1\} \qquad i,j \in V, k \in \mathcal{K} \tag{3.2h}$$

where (3.2b) are degree constraints, (3.2c) are the generalized subtour elimination constraints (GSECs), and (3.2d) fix the number of visits to a customer to be at most one. Constraints (3.2e)–(3.2g) determine the starting time of each route. Specifically, (3.2e) establish that a route cannot depart prior to the release date of each customer to be visited, (3.2f) fix the starting time of a route as the ending time of the preceding route, and (3.2g) set the ending time of the last route equal to T_E. Note that constraint (3.2g), as well as the equality in constraint (3.2f), is because the waiting time can be shifted to the beginning of the distribution without affecting the structure and the value of the solution, thus avoiding any waiting time between consecutive routes and at the end, as shown in Archetti et al. (2018). This way, the starting time of route $d_{|\mathcal{K}|}$ equals the ending time of the last route performed, $|\mathcal{K}| - 1$, which is, in turn, equal to the deadline T_E. We also add constraints (3.2i) and symmetry-breaking constraints (3.2j) to strengthen the formulation:

$$\sum_{i\in N} s_i^k \leq |N| s_0^k \qquad k \in \mathcal{K} \tag{3.2i}$$

$$s_0^k \leq s_0^{k+1} \qquad k \in \mathcal{K}, k \neq |\mathcal{K}| - 1. \tag{3.2j}$$

The optimal solution of (3.2) provides an upper bound to the DOP-rd. If the optimal solution is not found within a given computation time limit, the upper bound at the end of the computation (rounded down to the nearest integer) provides a valid upper bound for the DOP-rd. The model is solved using a branch-and-cut algorithm, in which the GSECs are separated on the fly. For more details on the separation of these constraints, see e.g., Lucena & Resende (2004), Ljubić (2021).

Note that Formulation (3.2) needs an upper bound on the number of routes $|\mathcal{K}|$ as input. A trivial upper bound is $|\mathcal{K}| = |N|$, which is the one used in Archetti et al. (2018). However, a tighter value can be obtained by solving the following ILP problem. Let us order customers $i \in N$ in increasing order according to the values of release dates (ties are broken arbitrarily). We define α_i as a binary variable equal to 1 if a direct route, from the depot to customer i and back, is performed, $i \in N$. A valid upper bound on the number of routes is given by the optimal solution of:

$$\max \sum_{i \in N} \alpha_i \tag{3.3a}$$

$$\text{subject to: } \alpha_i r_i + \sum_{j=i}^{j=|N|} (d_{0j} + d_{j0})\alpha_j \leq T_E \qquad i \in N \tag{3.3b}$$

$$\alpha_i \in \{0,1\} \qquad i \in N \tag{3.3c}$$

Formulation (3.3) counts the maximum number of direct routes that can be performed within T_E. Let us call UB_{route} the optimal solution of (3.3).

Proposition 1. *Given a vector of deterministic release dates r_i and deadline T_E:*

1. *UB_{route} is a valid upper bound on the maximum number of routes $|\mathcal{K}|$ for the OP-rd;*
2. *Any optimal solution of (3.3) is a feasible solution for (3.2).*

Proof. A solution of (3.2) consists of a sequence of non-empty feasible and compatible routes. A route is feasible if it does not start before any of the release dates of the customers to be visited and finishes before T_E. Routes are compatible if they visit each customer at most once. Considering a customer i, the shortest feasible route serving i is the one that leaves the depot at time r_i, goes to i and comes back to the depot. Constraints (3.3b) guarantee the feasibility of the direct routes selected: indeed, for each customer i, constraints (3.3b) ensure that the route to i, as well as the following direct routes can be performed before the deadline T_E. Thus, Formulation (3.3) determines the maximum number of direct routes performed within T_E. Let us assume that there exists a solution $\tilde{s}$ with a higher number of routes than UB_{routes}. Given the argument above, at least one of the routes in $\tilde{s}$ is not a direct route. Thus, we can arbitrarily remove customers from this route up to when a single customer remains, making it a direct route. This procedure can be repeated on all routes containing more than one customer in $\tilde{s}$. In this way, we obtain a solution with the same number of routes as in $\tilde{s}$, and where all routes are direct routes. However, due to the argument above, the number of routes cannot be greater than UB_{routes}, and this proves point 1 of the theorem. As for point 2, feasibility is ensured by constraints (3.3b): a route starts not earlier than the delivery date of the customer to be served, and its duration plus the duration of the following routes does not exceed T_E. □

3.4.2 Partial Characterization of Optimal Policy

The following proposition provides sufficient conditions to partially characterize an optimal policy.

Proposition 2.

Let ℓ be the largest number of parcels from $N_e^{unserved}$ that can be delivered within $[t_e, T_E]$. If there exists a subset $N_{sol} \subseteq N_e^{known}$ of parcels that can be delivered within $[t_e, T_E]$ such that $|N_{sol}| = \ell$, then the optimal policy is to leave the depot immediately and distribute the parcels from N_{sol}.

Proof. Proof. Trivial.

Hence, Proposition 2 provides sufficient conditions under which it is optimal to leave the depot and (partially) distribute the available parcels. Checking these conditions requires solving an NP-hard optimization problem. Indeed, given a decision epoch e, and the set of parcels from $N_e^{unserved}$, one has to find an optimal solution of the OP with respect to $N_e^{unserved}$ with a maximum route duration equal to $T_E - t_e$ and with all parcel profits set to one. In case the value of the optimal solution is $\ell > |N_e^{known}|$, then it immediately follows that the condition of Proposition 2 is not satisfied. Otherwise, one has to solve a new OP on the set N_e^{known}, and the condition of Proposition 2 holds if and only if the value ℓ of the optimal solution remains unchanged. Note that, in preliminary experiments, we also implemented a different approach for verifying Proposition 2, where a single OP was solved on $N_e^{unserved}$ where profits were set to 1 for parcels in $N_e^{unknown}$, and to $1 - \varepsilon$ for parcels in N_e^{known}, with a sufficiently small value of ε. However, the former approach turned out to be much more efficient than the latter one. Finally, we note that, for small values of $|N_e^{known}|$ and large values of $[t_e, T_e]$, it is unlikely that Proposition 2 holds, and in this case, we skip checking whether the underlying conditions hold (see Section 3.6 for more implementation details).

3.5 Reinforcement learning solution approaches

The solution approaches are based on Monte-Carlo simulation, where scenario generation is performed at each decision epoch e according to the state S_e, and on an approximation of the future routes, described in Section 3.5.1. Thus, an online approximation scheme is developed, which learns from updated information.

According to formula (3.1), in each decision epoch, the goal is to find an action $\mathcal{X}_e$, i.e., the "route 0" (the route that starts now), which maximizes the sum of the immediate reward $C(S_e, \mathcal{X}_e)$, corresponding to the number of parcels delivered in the route 0, and the expected number of parcels delivered by future routes. To anticipate future routes, at each decision epoch e, we generate a set of scenarios Ω associated with unserved customers predicting their release dates $\tilde{r}_e$. For each scenario, we approximate future routes using the batch approach described in Section 3.5.1. This approximation is used in both reinforcement learning approaches to determine the decision-making

policy, which are:

- *Value Function Approximation with a Consensus Function (VFA-CF):* for each scenario, we run a deterministic ILP model to determine the set of known requests that should be served with a route starting immediately and the set of requests included in future routes. Then, a consensus function based on the solutions obtained over all scenarios defines the set of known requests served immediately (if any), i.e., with a route leaving from the depot at time t_e.
- *Value Function Approximation with a Two-Stage Stochastic ILP Model (VFA-2S):* instead of looking at each scenario independently, we build a two-stage stochastic ILP model in which the first stage determines a set of requests to be delivered at time t_e, while the second stage concerns the expected number of parcels delivered by future routes by integrating all scenarios with equal probability.

As stated before, both methods are hybridized with a one-step look-ahead strategy. One-step look-ahead involves looking one step ahead and considering all possible actions that can be taken from the current state, and selecting the action that maximizes the immediate reward plus the expected reward of the next state. VFA involves estimating the value function for each state. We combine these two techniques to exert their advantages in the following way. We approximate the future impact of the current action rather than simulating all possible actions at the next state. At each decision epoch, we first approximate future routes using a batch approach. We then examine the available actions in the current state and evaluate each action's immediate reward (i.e., the number of parcels delivered by a route leaving now) and the next state's expected reward (i.e., the expected number of parcels delivered by future routes). Finally, we select the action that maximizes the sum of the immediate reward and the expected future reward.

As decisions for the DOP-rd must be taken quickly in practical contexts (e.g., in seconds/minutes), the considered approximation of future routes can be relevant to substantially speeding up computation and making the

approach applicable in practice. Without the approximation, using the VFA-CF/VFA-2S with exact methods to determine the entire sequence of future routes would be intractable.

The two approaches are described in Sections 3.5.2 and 3.5.3, respectively (an overview of additional notation used is provided in Table 3.3). Both ILP methods are based on state-of-the-art branch-and-cut techniques. Note that, in both approaches, Proposition 2 is checked at each decision epoch (before solving the corresponding ILP) and the procedure is stopped in case it is satisfied, as the best policy is identified by the OP solution defined by the proposition.

3.5.1 The Batch Approach

At each decision epoch e, we are given the sets of known customers N_e^{known} and unknown customers $N_e^{unknown}$. As for $N_e^{unknown}$, we generate a set of scenarios Ω where a scenario $\omega \in \Omega$ represents a possible realization of the release dates for $i \in N_e^{unknown}$, sampled according to the distribution associated with $\tilde{r}_i^e$. Assume that we are given a scenario $\omega \in \Omega$, we propose to determine the route serving a subset of requests from N_e^{known} (the route 0), while approximating the future routes and the respective rewards obtained from serving the remaining ones from $N_e^{unserved}$ until the deadline. We assume that there will be $|K|$ additional future routes serving customers in $N_e^{unserved}$, and for their approximation we use a batch approach. The term "batch" refers to a set of requests scheduled for future delivery, with each batch being served by a dedicated future route. On the other hand, the "batch approach" refers to the specific approach employed to create batches.

First, we assume that each future route will serve at most ρ requests. This will allow us to obtain an approximation of the duration of future routes easily. Also, it is reasonable to assume that in practical applications, the company will not wait too long (i.e., it will not wait for more than ρ parcels to become available at the depot) before dispatching the vehicle. Thus, by properly tuning the value of ρ (as shown in Section 3.6.2), one might obtain a

Table 3.3: Overview of notation used in solution approaches

Solution Approaches	
Ω	Set of scenarios predicting their release dates $\widetilde{r}_i^e$ at decision epoch e, associated with each customer $i \in N_e^{unserved}$
ω	A realization of possible values of release dates for each customer $i \in N_e^{unserved}$, $\omega \in \Omega$
ρ	Maximum number of requests to be served in each future route
$\mathcal{A}$	Euclidean area containing the locations of unserved customers
T_D	Expected tour duration
θ	Tuning parameters including T_D, the distribution of release dates, ρ, and $\lvert\Omega\rvert$.
$\tilde{S}_{e+1}$	Approximation of state at $e+1$
$\tilde{V}$	The value of the value function estimated
τ_{start}^0	A parameter indicating the starting time of the immediate route, so its value actually equals the time of the epoch e.
A^{known}	Set of arcs (i,j) where $i,j \in N_e^{known} \cup \{0\}$, linking customers with known requests and depot
p_ω	Probability associated with each scenario $\omega \in \Omega$
route 0	the first route leaving for delivery in $\mathcal{K}$, and for VFA approaches it starts immediately if non-empty in each decision epoch
Batch Approach	
K	Set of indices for batches or future routes
K_0	Set of indices for batches with spare capacities ($K_0 \subseteq K$)
$k(i)$	Index of the batch in which the request $i \in N_e^{unknown}$ should be served
τ_{start}^k	Starting time of the route serving batch $k \in K$
τ_{end}^k	Ending time of the route serving batch $k \in K$
ρ_k	Number of unknown requests assigned to the route serving batch $k \in K$
Value Function Approximation with a Consensus Function	
$\tilde{X}_e^\omega$	Optimal action for the given value function approximation $\tilde{V}$ under scenario $\omega \in \Omega$ in decision epoch e
Φ	Consensus function for deciding the solution under the consideration of all the scenarios
$\tilde{X}_e^{PFA}$	Action decided by the consensus function Φ
x_{ij}^0	Binary arc variable for route 0, equal to 1 if the route traverses arc $(i,j) \in A^{known}$, 0 otherwise
y_i^0	Binary variable associated with known requests $i \in N_e^{known}$, equal to 1, if customer i is visited in route 0, and 0 otherwise
z_k	Binary variable equal to 1 if future route $k \in K$ is executed, 0 otherwise
τ_{end}^0	Continuous variable denoting the ending time of route 0
π_i^k	Binary variable equal to 1 if known request $i \in N_e^{known}$ is served in batch $k \in K_0$, 0 otherwise
λ	the percentage of scenarios in which a location must appear for being included in route 0
Value Function Approximation with a Two-Stage Stochastic ILP Model	
K^ω	Set of indices for all the batches created in scenario ω, $\omega \in \Omega$
K_0^ω	Subset of batches with spare capacities in scenario $\omega \in \Omega$
$\tau_{start}^{k\omega}$	Starting time of batch $k \in K^\omega$ in scenario $\omega \in \Omega$
ρ_k^ω	Number of unknown requests assigned to the route serving batch $k \in K^\omega$ in scenario $\omega \in \Omega$
$\pi_i^{k\omega}$	Binary decision variable equal to 1 if known request $i \in N_e^{known}$ is served in batch k in scenario $\omega \in \Omega$, $k \in K_0^\omega$, and 0 otherwise
z_k^ω	Binary decision variable equal to 1 if batch $k \in K^\omega$ in scenario $\omega \in \Omega$ is executed, and 0 otherwise
$\widetilde{r}^e$	The distribution of uncertain release dates for unserved requests
$\tilde{V}(y^0, \tau_{end}^0, \widetilde{r}^e)$	The value of the recourse function associated with the second stage

good approximation of the duration of future routes. We apply the formula of Daganzo (1984) to estimate the duration of each future route. This formula is commonly used in literature thanks to the fact that it is easy to implement and provides good estimation (Jabali et al. (2012) and Van Heeswijk et al. (2019)). According to the formula, given the Euclidean area $\mathcal{A}$ containing the locations of unserved customers and ρ, under the assumption that the locations are scattered uniformly and independently, the expected tour duration T_D is $0.75\sqrt{\mathcal{A}\rho}$. This approach particularly fits the deliveries in densely populated areas, where dispatchers are assigned a restricted number of parcels in each delivery, thus route duration is rather stable and easy to estimate.

The input to the batch approach is given by the duration of a batch route T_D (calculated through Daganzo's formula mentioned above), the distribution of release dates, and ρ, the maximum number of requests served in each future route (represented by a batch). All these parameters, together with $|\Omega|$ (the number of scenarios sampled to approximate the future release dates), are referred to as θ in the following. Then, for each scenario $\omega \in \Omega$, the batch algorithm works as described in the following. For ease of reading, we omit the index ω in the notations.

The batch approach (whose pseudo code is given in Algorithm 1) starts by ordering the unserved requests according to their release dates associated with the realization of scenario ω (Step 3). Recall that $\widetilde{r}_i^e$ are random variables representing the release dates for $i \in N_e^{unserved}$. For a given scenario ω we will denote by r_i^e the realization drawn from the release date distribution for $i \in N_e^{unknown}$ and actual release dates for $i \in N_e^{known}$. The algorithm then divides requests into batches. To serve as many requests as possible, the last route (or batch) is supposed to serve the requests arriving at the latest that can still be feasibly served before the deadline. Thus in Step 5 the selection is made backwards, i.e., starting from the request with the latest release date, and checking whether it can be included in the last route, i.e., if its release date is not larger than the earliest possible starting time of the last route, which is $T_E - T_D$. If true, the number of requests q in the assigned batch is increased by one. Subsequently, if the request is known, the batch k is added to K_0; otherwise, request i is allocated to the batch k and the counter ρ_k is updated. After that, in Step 13, if the batch size limit is reached ($q = \rho$) or all requests are processed ($i = 1$), the start and end time of the batch are updated, and counters are reset (Steps 14-16). The algorithm iteratively

follows the procedure, checking the next sorted requests. Then a new route is created (the second last) associated with a starting time equal to $T_E - 2T_D$, and so on. In this manner, we continue checking the next requests until all are examined or the start time of a new batch t_k is less than or equal to the time at decision epoch e (Step 6). To be consistent with the earlier notation, route 0 refers to the route supposed to leave the depot immediately. We introduce the new notation for a detailed description of the algorithm:

- K: the set of indices for all the batches created (we use the same notation as the set of future routes since each batch will be served by a future route. However, the IDs are annotated reversely, i.e., batch 1 is served by the last route, batch 2 is served by the second last route, etc);
- K_0: the set of indices for batches with positive spare capacities ($K_0 \subseteq K$). We say that a batch k has a positive spare capacity if the number of requests in $N_e^{unknown}$ assigned to k is less than ρ;
- $k(i)$: the index of the batch request i is assigned to, $i \in N_e^{unknown}$;
- τ_{start}^k: the starting time of the route serving batch $k \in K$;
- τ_{end}^k: the ending time of the route serving batch $k \in K$.
- ρ_k: the number of requests in $N_e^{unknown}$ assigned to the route serving batch $k \in K$.

The scheme of the batch algorithm is presented in Algorithm 1.

Note that in Algorithm 1, customers in N_e^{known} are treated differently than those in $N_e^{unknown}$. For each customer $i \in N_e^{unknown}$ such that $r_i^e \leq T_E - T_D$ the algorithm determines the unique batch $k(i) \in K$ to which i is assigned. However, the customers in N_e^{known} can be assigned to any potential batch $k \in K$ with some spare capacities, i.e., which contains less than ρ unknown requests. The subsets of the batches with spare capacities are denoted by K_0, and a unique assignment of a request $i \in N_e^{known}$ to some $k \in K_0$ is obtained through the optimization models proposed in Sections 3.5.2 and 3.5.3.

Algorithm 1 The batch algorithm

Require: Time t_e; Unserved requests $N_e^{unserved} = N_e^{known} \cup N_e^{unknown}$; Release dates $\{r_i^e, i \in N_e^{unserved}\}$; Deadline T_E; Maximum number of requests allowed in a batch ρ; Duration of a batch route T_D;

Ensure: Set K_0; Number of batches $|K|$; Batch indices $\{k(i), i \in N_e^{unknown}\}$; $\{(\tau_{start}^k, \tau_{end}^k, \rho_k), k \in K\}$;

$n \leftarrow |N_e^{unserved}|$; $K \leftarrow \emptyset$; $K_0 \leftarrow \emptyset$; $k \leftarrow 1$; $t_k \leftarrow T_E$ - T_D; $\rho_1 \leftarrow 0$
$k(i) \leftarrow 0$, for all $i \in N_e^{unknown}$;
Sort the requests from $N_e^{unserved}$ in increasing order of release dates r_i^e;
Initialize the number of requests assigned to the k-th batch $q \leftarrow 0$;
for $i \leftarrow n$ to 1 **do**
    if $t_k \leq t_e$ **then break**
    if release date $r_i^e \leq t_k$ **then**
        $q \leftarrow q + 1$
        if $i \in N_e^{known}$ **then**
            if $k \notin K_0$ **then** $K_0 \leftarrow K_0 \cup \{k\}$
        else $k(i) \leftarrow k$; $\rho_k \leftarrow \rho_k + 1$
        end if
        if $q = \rho$ or $i = 1$ **then**
            $K \leftarrow K \cup \{k\}$
            $\tau_{start}^k \leftarrow t_k$; $\tau_{end}^k \leftarrow t_k + T_D$
            $k \leftarrow k + 1$; $t_k \leftarrow t_k - T_D$; $q \leftarrow 0$; $\rho_k \leftarrow 0$
        end if
    end if
end for
return $K_0, K, \{k(i), i \in N_e^{unknown}\}$, and $\{(\tau_{start}^k, \tau_{end}^k, \rho_k)\}_{k \in K}$

Table 3.4: Input instance for the batch approach (*release date (RD) 0 means the parcel i is at the depot, i.e., $i \in N_e^{known}$*)

Total number of requests	9	**Max number of requests in a batch** ρ							3
Number of batches	3	**Duration of a batch route** T_D							1
Returning time of route 0	TBD	**Deadline**							16
ID of the requests	1	2	3	4	5	6	7	8	9
RD of the requests	0	0	0	14	14	15	15	15	15

Table 3.5: The batches

Batch number	**3**	**2**			**1**		
Unknown requests assigned	-	-	4	5	7	8	9
Max RD served	0	14			15		
Start time of the tour	13	14			15		
End time of the tour	14	15			16		
k(i)	-	-	$k(4) = 2$	$k(5) = 2$	$k(7) = 1$	$k(8) = 1$	$k(9) = 1$
K_0	✓	✓			✗		

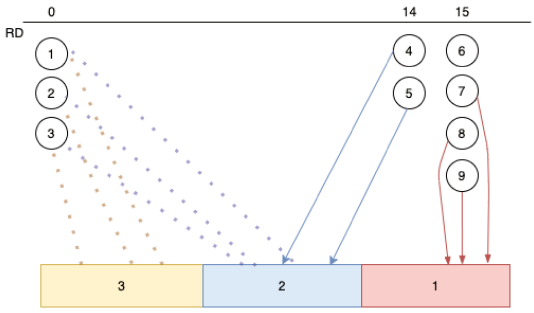


Figure 3.1: The assignment of requests to batches (*a dashed line with no arrow connects known requests with batches from K_0, indicating potential assignments of each $i \in N_e^{known}$ to some $k \in K_0$)*

To explain how the batch approach works, a toy example with 9 requests is displayed in Table 3.4, and the output is shown in Table 3.5 and Figure 3.1. Note that in Table 3.4 we also consider the ending time of route 0 (which is a decision variable defined in the ILP model, where it is denoted as τ^0_{end}, see Section 3.5.2.2), that is the route leaving immediately from the depot. Its ending time will allow scheduling only those batches that can start after route 0 is completed. The role of this parameter will become clearer in the following sections, where we explain how the output of the batch algorithm is integrated with the two solution approaches. As the deadline is 16, the first batch route should be completed no later than 16. Each route lasts one time unit (according to Daganzo's formula), so the tour's start time is 15. A batch can serve three requests at most. As the value of the solution does not depend on the subset of requests served but just on their number, we arbitrarily assign to batch 1 (which is mapped as the last future route) the three requests with the highest index. For batch 2, its ending time is 15, so its starting time is 14. Thus the two requests with sampled release dates equal to 14 are assigned to the batch. As batch 2 has a residual capacity of 1, any known requests (with IDs 1, 2, and 3) can be assigned to it. In addition, requests 1, 2, and 3 can be assigned to batch 3. Table 3.5 describes the content of batches and displays the values of $k(i)$. It also shows that batches 2 and 3 are the ones that can be used to serve known requests (thus, they belong to set K_0) with the check mark on the last row. If a request $i \in N_e^{unknown}$ is not assigned to any batch (for example, because its release date is larger than the starting time of batch 1 or because there are no sufficient batches to include all requests), then $k(i) = 0$.

3.5.1.1 Optimality Conditions for the Batch Approach

In the following, we show that under some simplifying conditions, the batch approach given by Algorithm 1 provides a polynomial way to calculate an optimal solution. Let us assume that we are at the decision epoch e and that all the release dates are known with certainty. We denote as r the vector of release dates. In the following proposition, we still use the notation $N_e^{unknown}$ referring to parcels whose release date is greater than t_e.

Proposition 3. *If the release dates are deterministic, the duration of each route is T_D, independently of the location of the requests served, and the vehicle can deliver up to ρ parcels per route, Algorithm 1 finds (in polynomial time):*

1. *The maximum number of requests from $N_e^{unknown}$ that can be served within $[t_e, T_E]$, given as $|N_{\mathcal{O}}|$, where $N_{\mathcal{O}} = \{i \in N_e^{unknown} : k(i) \neq 0\}$.*
2. *The maximum number of all requests from $N_e^{unserved}$ that can be served within the interval $[t_e, T_E]$, given as:*

$$OPT_{\rho,T_D} = \begin{cases} |N_e^{known}| + |N_{\mathcal{O}}|, & \text{if } |N_e^{known}| < \sum_{k \in K_0} (\rho - \rho_k) \\ |K| * \rho, & \text{otherwise} \end{cases}$$

Proof. We first observe that in case $\rho_k = \rho$ for all $k \in K$, then this is trivially the optimal solution. Thus, in the following, we discard this case.

1. By construction of the solution $N_{\mathcal{O}}$, there exists $\tilde{k}$, $1 \leq \tilde{k} \leq |K|$, such that $\rho_{\tilde{k}} < \rho, \rho_k = \rho$ for $k < \tilde{k}$ and $\rho_k = 0$ for $k > \tilde{k}$. The batch $\tilde{k}$ is called *critical batch* and corresponds to the only possible batch whose spare capacity is positive but strictly smaller than ρ. We call this property *the critical batch property*. Suppose that there exists a solution $N_{\mathcal{O}'} \subseteq N_e^{unknown}$ such that $|N_{\mathcal{O}'}| > |N_{\mathcal{O}}|$ parcels from $N_e^{unknown}$ can be delivered within the interval $[t_e, T_E]$. Without loss of generality, we can shift the starting time of all batches to the latest possible and then reassign parcels to the latest possible batch so that the solution $N_{\mathcal{O}'}$ also satisfies the

critical batch property. We argue that the number of parcels scheduled in batch $\tilde{k}$ of the reordered solution $N_{\mathcal{O}'}$ cannot be bigger than the one in $N_{\mathcal{O}}$. Indeed, by contradiction: let us consider a request i served in batch $\tilde{k}$ in $N_{\mathcal{O}'}$ and not in $N_{\mathcal{O}}$. In case i is not served in any batch $k < \tilde{k}$ in $N_{\mathcal{O}}$, then i would have been inserted in batch $\tilde{k}$ in solution $N_{\mathcal{O}}$ as well. In case, instead, request i is served in one batch $k < \tilde{k}$ in $N_{\mathcal{O}}$, then it means there exists at least one request i' served in a batch $k < \tilde{k}$ in $N_{\mathcal{O}'}$ but not in $N_{\mathcal{O}}$. However, due to the way batches in $N_{\mathcal{O}}$ are constructed, this means that $r_i^e \geq r_{i'}^e$. Given that i is inserted in batch $\tilde{k}$ in $N_{\mathcal{O}'}$, this means that $r_{i'}^e \leq \tau_{start}^{\tilde{k}}$, thus i' and i could be swapped in the corresponding batches in $N_{\mathcal{O}'}$. By repeating iteratively this procedure on all requests in batch $\tilde{k}$ in $N_{\mathcal{O}'}$, we obtain two identical sets of requests in the critical batch $\tilde{k}$.

2. We notice that $\sum_{k \in K_0}(\rho - \rho_k)$ represents the sum of spare capacities over all routes that can be scheduled within $[t_e, T_E]$. Since the parcels from N_e^{known} are already available at time t_e, if their number is smaller than the total spare capacity, all of them can be distributed among the routes determined by K_0, hence guaranteeing that $|N_e^{known}| + |N_{\mathcal{O}}|$ parcels are delivered in total. If, on the contrary, $|N_e^{known}|$ is larger than the total spare capacity, only a subset of them will be delivered so that for each batch, its maximal capacity is used. We emphasize that due to the assumption that all routes have the same duration, namely T_D, and the same capacity ρ, customer locations are irrelevant in this case, and hence, the way how assignment of parcels from N_e^{known} to batches from K_0 is done, does not affect the optimal solution. □

Hence, by combining scenario generation and the batch approach, we approximate the value of future routes as a one-step look-ahead policy in which future events are condensed in a single step, and future actions are evaluated through the batch approach.

Finally, to evaluate if each future route is executable, we need to check whether the starting time is greater than the ending time of the route leaving immediately from the depot, that is route 0. This is done through the optimization models presented in the following section.

3.5.2 Value Function Approximation with a Consensus Function (VFA-CF)

There are four fundamental ways of deriving policies in MDPs (Powell, 2014), one of which is based on value function approximations (VFAs) where the whole term representing the value function of the future states in (3.1) is replaced by an approximation. Our solution approaches fall into this category. VFA-CF relies on the batch approach to approximate future routes, and uses a consensus function working according to the following rule:

$$\tilde{X}_e^{VFA-CF}(S_e \mid \theta) = \Phi(\tilde{X}_e^{\omega})_{\omega \in \Omega} \tag{3.4}$$

$$\text{where} \qquad \tilde{X}_e^{\omega} = \arg \max_{\mathcal{X}_e \in X(S_e)} \{C(S_e, \mathcal{X}_e) + \gamma \tilde{V}_{e+1}(\tilde{S}_{e+1} \mid S_e, \mathcal{X}_e, \omega)\} \tag{3.5}$$

is the optimal action for the given value function approximation $\tilde{V}_{e+1}(\tilde{S}_{e+1} \mid S_e, \mathcal{X}_e, \omega)$ ($\tilde{S}_{e+1}$ is the approximation of state at $e+1$ and $\tilde{V}$ is an estimate of the value of being in state $\tilde{S}_{e+1}$), under the realization of scenario ω. As defined in Section 3.5.1, θ captures all tunable parameters of the model. Function Φ is the consensus function for determining the action, considering the best actions over all scenarios (see Section 3.5.2.1). Thus, the main idea of the proposed VFA-CF is that for each scenario ω, the best action $\tilde{X}_e^{\omega}$ is determined, according to the method described in Section 3.5.2.2. This consists in solving a deterministic ILP where the goal is to determine the route that should leave immediately from the depot (which might be empty) by optimizing the function given by the sum of the immediate reward associated with this route plus the estimated number of requests to be served in future routes (determined through the batch approach described in Section 3.5.1). Then, a consensus function is applied to determine the best action according to the solution obtained on each of the scenarios. Note that, as we solve an ILP for each scenario independently, there is no uncertainty in the input data to the ILP: indeed, the release dates correspond to the realizations associated with the given scenario, and we refer to this model as *a deterministic ILP model* below.

3.5.2.1 Consensus Function

We define the consensus function, denoted by Φ in formula (3.4), as the policy to determine which requests are served in route 0 while considering all solutions associated with different scenarios. The scheme is reported in Algorithm 2. The function checks solutions $\tilde{X}_e^\omega$ over all scenarios $\omega \in \Omega$, by focusing on the requests in N_e^{known} served in route 0 in each scenario. The idea is to determine which subset of these requests should be served in route 0 at the current decision epoch. Specifically, a request $i \in N_e^{known}$ is served if it is included in route 0 in at least $\lceil \lambda |\Omega| \rceil$ scenarios, with $0 < \lambda < 1$. If at least one request is included in the route, then a Traveling Salesman Problem (TSP) is solved over all requests included to determine the best sequence, thus determining the values of the arc variables $x_{ij}^e, i, j \in N_e^{known} \cup \{0\}$ associated with the action $\mathcal{X}_e$. If the duration of the TSP exceeds T_E, an OP model is solved to determine the requests to be served, followed by the solution of TSP over the selected requests to get the best visit sequence. Otherwise, the decision is to wait at the depot until the next decision epoch $e + 1$.

3.5.2.2 Deterministic ILP Model

We now describe the optimization model used to find the decision $\tilde{X}_e^\omega$ given in (3.5). The idea is that only the reward of route 0 (which corresponds to $C(S_e, \mathcal{X}_e)$) is calculated with the detailed routing information (since the route is to be executed immediately in the case at least one request is assigned to it), while the value and the duration of future routes are estimated through the batch approach (Section 3.5.1). We define as $K = \{1, ..., |K|\}$ the set of indices for all the batches created where each batch is served by a future route. The parameters considered are the following:

- τ_{start}^0, a parameter indicating the starting time of the immediate route, so its value equals t_e.
- A^{known}, it includes all arcs (i, j) where $i, j \in N_e^{known} \cup \{0\}$ linking customers with known requests and depot.

Algorithm 2 Consensus Function

Require: Time t_e; N_e^{known}; a set of scenarios Ω; solution $\tilde{X}_e^{\omega}$ of scenario ω;
Ensure: a route R;

```
Initialize the frequency of all requests: F_e(i) ← 0 i ∈ N_e^known; Initialize the set of requests to be served as I_sol ← ∅;
for ω ∈ Ω do
    I_ω ← the set of requests served in route 0 of the solution X̃_e^ω of scenario ω.
    for i ∈ I_ω do F_e(i) ← F_e(i) + 1
    end for
end for
for i ∈ N_e^known do
    if F_e(i) ≥ λ|Ω| then I_sol ← I_sol ∪ {i}
end for
R ← TSP over I_sol
if length of R > T_E − t_e then
    I_sol ← OP over I_sol
    return R ← TSP over I_sol
else
    return R
end if
```

In addition, we have the parameters associated with future batches corresponding to the output of the batch approach (see Section 3.5.1). The decision variables are the following:

- x_{ij}^0: binary arc variable for route 0, equal to 1 if the route traverses arc $(i,j) \in A^{known}$, 0 otherwise.
- y_i^0: binary variable associated with known request $i \in N_e^{known}$, equal to 1 if customer i is served in route 0, 0 otherwise.
- z_k: binary variable equal to 1 if future batch $k \in K$ is executed, 0 otherwise.
- τ_{end}^0: continuous variable denoting the ending time of route 0.
- π_i^k: binary variable equal to 1 if $i \in N_e^{known}$ is served in batch $k \in K_0$, 0 otherwise.

The deterministic ILP model is given as follows. In line with (3.5), the first term in the objective function (3.6a), namely $\sum_{i \in N_e^{known}} y_i^0$, maps $C(S_e, \mathcal{X}_e)$ representing the number of requests served by action $\mathcal{X}_e$. The sum of the two remaining terms in (3.6a), namely $\sum_{k \in K} \rho_k z_k$ and $\sum_{k \in K_0} \sum_{i \in N_e^{known}} \pi_i^k$ maps $\tilde{V}_{e+1}(\tilde{S}_{e+1} \mid S_e, \mathcal{X}_e, \omega)$, representing the approximated number of parcels delivered by future routes. Note that in case $\sum_{i \in N_e^{known}} y_i^0 = 0$, it means that route o does not serve any request and the decision is to wait at the depot until the next decision epoch $e+1$. Then for each scenario ω, the corresponding ILP formulation is given as follows:

$$\max \sum_{i \in N_e^{known}} y_i^0 + \gamma \left[\sum_{k \in K} \rho_k z_k + \sum_{k \in K_0} \sum_{i \in N_e^{known}} \pi_i^k \right] \tag{3.6a}$$

$$\text{subject to:} \sum_{(i,j) \in A^{known}} x_{ij}^0 = \sum_{(j,i) \in A^{known}} x_{ji}^0 = y_i^0 \qquad i \in N_e^{known} \cup \{0\} \tag{3.6b}$$

$$\sum_{i,j \in S} x_{ij}^0 \leq \sum_{i \in S \setminus \ell} y_i^0 \qquad S \subseteq N_e^{known}, |S| \geq 2, \ell \in S \tag{3.6c}$$

$$\tau_{end}^0 = \tau_{start}^0 + \sum_{(i,j) \in A^{known}} d_{ij} x_{ij}^0 \tag{3.6d}$$

$$\tau_{end}^0 \leq T_E \tag{3.6e}$$

$$\sum_{k \in K_0} \pi_i^k + y_i^0 \leq 1 \qquad i \in N_e^{known} \tag{3.6f}$$

$$\sum_{i \in N_e^{known}} \pi_i^k \leq (\rho - \rho_k) z_k \qquad k \in K_0 \tag{3.6g}$$

$$z_{k+1} \leq z_k \qquad k = 1, \ldots |K| - 1 \tag{3.6h}$$

$$\tau_{end}^0 - \tau_{start}^k \leq (T_E - \tau_{start}^k)(1 - z_k) \qquad k \in K \tag{3.6i}$$

$$y_i^0, \pi_i^k, x_{ij}^0, z_k \in \{0, 1\} \qquad i, j \in N_e^{known}, k \in K \tag{3.6j}$$

Besides, we initialize the model with GSECs of size two and the constraints (3.6l) to strengthen the formulation:

$$x^0_{ij} + x^0_{ji} \leq y^0_i \qquad i,j \in N_e^{known} \tag{3.6k}$$

$$\sum_{i \in N_e^{known}} y^0_i \leq |N| y^0_0 \tag{3.6l}$$

The objective function (3.6a) maximizes the total number of requests served by route 0 plus the approximated number of requests served by future routes, distinguished in unknown and known requests, multiplied by the discount factor γ. Constraints (3.6b), (3.6c) guarantee that route 0 is a circuit connected to the depot and prevent generating subtours. Constraint (3.6d) determines the ending time of route 0, which is computed as its starting time plus the traveling time. Constraints (3.6e) ensure that route 0 finishes before the deadline T_E. Constraints (3.6f) state that the same request cannot be served by both route 0 and any future route serving a batch in K_0, which guarantees the unique assignment of each known request as mentioned in Section 3.5.1. Constraints (3.6g) guarantee that each future route serving batches in K_0 does not serve more than ρ requests. Note that these constraints are needed for batches in K_0 only, as any request in N_e^{known} can potentially be assigned to each of these batches. Instead, for the batches in $K \backslash K_0$, the constraint is satisfied by construction from the batch approach. Constraints (3.6h) state that if batch $k+1$ is executed, batch k must be executed too. Constraints (3.6i) ensure that the future route serving batch k can start only after finishing route 0. Finally, (3.6j) define the variables domain.

To solve this model, we implemented a branch-and-cut procedure in which GSECs (3.6c) are separated in each node of the branching tree (see Section 3.6 for further details).

3.5.3 Value Function Approximation with a Two-Stage Stochastic ILP Model (VFA-2S)

The second approach we propose is a two-stage stochastic ILP model using a VFA:

$$\tilde{X}_e^{VFA-2S}(S_e \mid \theta) = \arg \max_{\mathcal{X}_e \in X(S_e)} \{C(S_e, \mathcal{X}_e) + \gamma \mathbb{E}[\widetilde{V}_{e+1}(\widetilde{S}_{e+1} \mid S_e, \mathcal{X}_e)]\} \tag{3.7}$$

where, as before, θ captures all tuning parameters, $\widetilde{S}_{e+1}$ is the approximation of state at the epoch $e+1$, and $\widetilde{V}_{e+1}(\widetilde{S}_{e+1})$ represents the approximation of the value function of being in state $\widetilde{S}_{e+1}$. As VFA-CF, it belongs to VFA since it also approximates the future impact (cost-to-go, or value function) of the current action (route 0) using the future routes returned by the batch approach, which will be explained in the following.

The action $\tilde{X}_e^{VFA-2S}(S_e \mid \theta)$ is determined through the solution of a two-stage stochastic ILP model as presented in the following.

3.5.3.1 Two-Stage Stochastic ILP Model

The modeling idea is similar to the one presented in Section 3.5.2.2, i.e., routing decisions are taken only for route 0 while future routes are approximated with the batch approach. The main difference with respect to the approach presented in Section 3.5.2 is that a single Two-Stage Stochastic Integer Linear Programming Model (2-SIP) is solved using all scenarios in Ω, instead of solving $|\Omega|$ deterministic ILPs (separately per scenario). Specifically, each scenario $\omega \in \Omega$ is associated with the realization of the release dates of the unknown requests $\{\widetilde{r}_i^e\}_{i \in N_e^{unknown}}$, denoted as $\widetilde{r}^{e1}, ..., \widetilde{r}^{e\omega}$. Also, we denote as p_ω ($0 < p_\omega < 1$) the probability associated with scenario $\omega \in \Omega$ ($\sum_{\omega \in \Omega} p_\omega = 1$). We first run the batch algorithm on the realization of each scenario to pre-calculate the following parameters:

- K^{ω} is the set of indices for all the batches created in scenario ω, $\omega \in \Omega$,
- K_0^{ω} is the subset of batches with spare capacities in scenario $\omega \in \Omega$,
- $\tau_{start}^{k\omega}$ is the starting time of batch $k \in K^{\omega}$ in scenario $\omega \in \Omega$,
- ρ_k^{ω} is the number of unknown requests assigned to batch $k \in K^{\omega}$ in scenario $\omega \in \Omega$.

As for the decision variables, we keep variables x_{ij}^0, y_i^0 and τ_{end}^0 as defined in the deterministic model (3.6). In addition, we have the scenario-related variables as follows:

- $\pi_i^{k\omega}$: $i \in N_e^{known}, k \in K_0^{\omega}$, binary variable equal to 1 if known request i is served in batch k in scenario $\omega \in \Omega$, and 0 otherwise.
- z_k^{ω}: Binary variable equal to 1 if batch $k \in K^{\omega}$ in scenario $\omega \in \Omega$ is used, and 0 otherwise. For ease of reading, we will simply write z_k^{ω} instead of $z_{k^{\omega}}^{\omega}$.

The deterministic equivalent of the 2-SIP is given by model (3.8). The objective function is:

$$\max \sum_{i \in N_e^{known}} y_i^0 + \gamma \sum_{\omega \in \Omega} p_{\omega} \Big(\sum_{k \in K^{\omega}} \rho_k^{\omega} z_k^{\omega} + \sum_{k \in K_0^{\omega}} \sum_{i \in N_e^{known}} \pi_i^{k\omega} \Big) \tag{3.8a}$$

The first-stage decision corresponds to determining route 0. In line with the function (3.7), the number of requests served by route 0 is determined by $\sum_{i \in N_e^{known}} y_i^0$ and matches the immediate reward $C(S_e, X_e)$. The second-stage decisions indexed by ω are associated with determining the requests served in future routes of scenario ω. Hence the second term in the objective function represents the expected value of future states with respect to the set of scenarios Ω. The constraints are:

$$\text{s.t.} \qquad (x_{ij}^0, y_i^0, \tau_{end}^0) \text{ satisfies (3.6b)-(3.6e), (3.6k)-(3.6l)}$$

$$\sum_{k \in K_0^\omega} \pi_i^{k\omega} \leq 1 - y_i^0 \qquad \omega \in \Omega, i \in N_e^{known} \tag{3.8b}$$

$$\sum_{i \in N_e^{known}} \pi_i^{k\omega} \leq (\rho - \rho_k^\omega) z_k^\omega \qquad \omega \in \Omega, k \in K_0^\omega \tag{3.8c}$$

$$z_k^\omega \geq z_{k+1}^\omega \qquad \omega \in \Omega, k = 1, \dots |K^\omega| - 1 \tag{3.8d}$$

$$\tau_{start}^{k\omega} + (T_E - \tau_{start}^{k\omega})(1 - z_k^\omega) \geq \tau_{end}^0 \qquad \omega \in \Omega, k \in K^\omega \tag{3.8e}$$

$$\pi_i^{k\omega}, z_k^\omega \in \{0,1\} \qquad \omega \in \Omega, i \in N_e^{known}, k \in K^\omega \tag{3.8f}$$

$$x_{ij}^0, y_i^0 \in \{0,1\} \qquad i,j \in N_e^{known} \tag{3.8g}$$

The objective function (3.8a) maximizes the total number of requests served by route 0 plus the expected number of requests served by future routes. The constraints (3.8b) - (3.8e) have a similar meaning as constraints (3.6f)-(3.6i) of Formulation (3.6), with the difference that they are replicated over all scenarios. Finally, (3.8f) and (3.8g) define the variables domain.

In order to solve this 2-SIP model, we implement a branch-and-cut procedure, similar to the one of the deterministic ILP model, with a dynamic separation of GSECs (3.6c). If at least one request is included in such obtained route 0, then a TSP is solved over all requests included to determine the best sequence formed by arcs x_{ij}^e, $i,j \in N_e^{known} \cup \{0\}$.

3.6 Computational experiments

In this section, we describe the computational experiments we carried out to validate the performance of the two proposed approaches. The code is developed in Python 3.7 and CPLEX version 20.1.0.0 is used to solve all ILPs, run on a single thread with a memory usage limit set to 2GB. We run the simulations on HP Z4 G4 Workstation with Intel(R) Xeon(R) W-2255 CPU @ 3.70GHz.

The maximum time interval between consecutive decision epochs φ is set to 10, which is a sufficiently small value to avoid long waiting times at the depot according to the input instances used in the computational campaign (described in Section 3.6.1).

As for implementation details, we remark on the following. Seven branch-and-cut algorithms are implemented: a) to calculate upper bounds with perfect information, b) to check whether the conditions of the partially optimal policy are satisfied, c) to calculate the consensus function of the VFA-CF approach, d) to calculate the value function approximation in the VFA approaches (VFA-CF and VFA-2S), e) to get the route for the ME approach described in Section 3.6.3.1 and f) to calculate the heuristic solution based on the modified OP-rd described in Section 3.6.3.2. All these algorithms rely on the separations of GSECs, which are implemented using lazy cutcallback of the CPLEX solver. To benefit from general-purpose cuts generated by the CPLEX solver, we collect the cuts separated at the root node of the branching tree and then restart the ILP. We set a time limit for the solution of each branch-and-cut as follows: VFA-CF: 5 minutes, VFA-2S: 10 minutes, ME: 10 minutes, and OPrd_PE: 2 minutes. Note that the deterministic model is allocated with a shorter computing time as it is solved for each scenario. As for the OP-rd heuristic model, the maximum time assigned is set to 2 minutes to be consistent with a practical application setting where a short time is available to take decisions in every decision epoch. All other CPLEX parameters are left at their default values. The call to the partial characterization of the optimal policy is skipped in each decision epoch of VFA-CF and VFA-2S when $|N_e^{known}| < 0.25|N_e^{unserved}|$ and remaining time $[t_e, T_E] > 0.75T_E$. We consider that for such settings there are small chances that conditions of Proposition 2 are satisfied. Concerning the calculation of upper bounds using model (3.2), we provide CPLEX with a starting feasible solution corresponding to the solution of model (3.3), and the time limit is set to one hour.

The remainder of this section is organized as follows. We first describe the input instances in Section 3.6.1 and the tuning of hyper-parameters in Section 3.6.2. Then in Section 3.6.3 we present the benchmark approaches we use for comparisons including two myopic approaches and an approach considering point estimation of future information. In Section 3.6.4, we first illustrate the benefit of integrating partial characterization of optimal policy

within VFA-CF/VFA-2S in Section 3.6.4.1. Given the benefits of PC and the fact that computing time is reduced or remains reasonable, we later show the results of VFA-CF/VFA-2S with PC in Section 3.6.4.2 by comparing them with the benchmark approaches, including a scalability test with 100 customers, and then we extend the comparison with a deterministic and static approach in Section 3.6.4.3 and the experiments for two vehicles in Section 3.6.4.4.

3.6.1 Input Instances

We use the set of benchmark instances described in Archetti et al. (2020a) that are derived from Solomon's instances (Solomon, 1987). These instances are publicly available at Archetti et al. (2020b). Solomon's instances contain six sets of instances: $C1$, $C2$, $R1$, $R2$, $RC1$, and $RC2$. The instances in the same set share the same vertex coordinates and vary in time windows only. Because we do not discuss time windows, we keep one instance from each set. Besides, we focus on $C1$, $C2$, $R1$, and $RC1$, excluding $R2$ and $RC2$ due to their identical coordinates with $R1$ and $RC1$, respectively. Each of the instances is annotated as I_β, I is the instance name, including $C101$, $C201$, $R101$, and $RC101$, where the vertex coordinates are taken, and β describes how much the release dates spread out, compared to the TSP travel time of the instance.

The release dates are generated using a Gaussian distribution with the expectation and the variance obtained by simulating the arrival of the vehicles transporting the packages to the depot. For customers in $N_e^{dynamic}$ (i.e., the set of customers whose information about the release date is dynamically updated along with the parcels' travel to the depot), both the expectation and the variance are updated by simulating the traveling of the vehicle for delivering the packages to the depot. The distance of each vehicle is updated by reducing the distance traveled in the previous time unit. If the updated distance is nonpositive at e, it means the vehicle has delivered the parcel to the depot, and the customer i is added in the set N_e^{known} with release date $r_i = t_e$. Otherwise, the vehicle speed is updated as a truncated random walk process with Gaussian steps, and the new expectations and variances of the distributions

are computed as well. Specifically, as the vehicle travels to the depot, the uncertainties decrease, and the estimation of the release date gets more precise. We use $\delta \in \{0, 0.5, 1\}$ to indicate the rate of customers with a dynamically updated release date. When δ is 0, it means all distributions of customers' release dates are static. Besides, for each Solomon's instance, a parameter $\beta \in \{0.5, 1, 1.5\}$ is generated, defining the dispersion of release dates. The greater the value of β, the broader the interval the release dates are sampled from.

Each instance contains 50 customers and is associated with three parameters β, δ, and T_E. Parameter T_E is the deadline of the delivery service (not defined in the problem studied in Archetti et al. (2020a)), which is determined as follows. We first determine the latest actual arrival time of all requests and denote it as $T_{standard}$. Then we generate four instances with T_E equal to $c\ T_{standard}$, where $c \in \{0.6, 0.8, 1.0, 1.2\}$. For each customer input data and each choice of β, δ, and c, five different instances (with different seeds) are generated, in which two are used for hyper-parameter tuning (288 instances), and three are used for the final tests (432 instances). As for the pairwise Euclidean distances between customer coordinates, they are rounded up to the lowest integer values. In our instances, we have some parcels available in the beginning, and they could be dispatched immediately (if necessary). To test the scalability of the proposed approaches, we created 36 additional instances containing 100 customers, where the customers' locations are generated by combining instances C101 and RC101, and dynamic release dates are generated by combining the ones of instances C101 and RC101 accordingly. The instances, along with a detailed description of the instance generation procedure, can be found at Li et al. (2023).

3.6.2 Hyper-parameter tuning

To justify the choice of our hyper-parameters, two out of the five seeds of instance generation have been used for hyper-parameter tuning. Specifically, we tune the number of scenarios $|\Omega|$, batch size ρ, discount factor γ, and λ - the percentage of scenarios in which a location must appear to be included in route 0. We run the policies with PC with different parameter values and present the results in the following. We adopt a set of default parameter

values: $|\Omega| = 30, \gamma = 0.7, \lambda = 0.5$. When we tune one of the parameters, we use the default values of the other parameters.

To compare the performance, we present the percentage gap from the best policy among the set of parameter values for the same approach. To better clarify, we provide an example of scenario numbers. For each of the two approaches $p \in \{\text{VFA-CF, VFA-2S}\}$, and a given input instance, let $N_{p,|\Omega|}$ denote the number of total requests served, assuming approach p is applied with a given value of $|\Omega|$. Then

$$\Gamma_{p,|\Omega|} = 1 - \frac{N_{p,|\Omega|}}{\max_{|\Omega| \in \{10,30,50\}} N_{p,|\Omega|}} \tag{3.9}$$

provides the gap between the policy obtained for a given value of $|\Omega|$ and the best policy among the three different values of $|\Omega|$.

3.6.2.1 Number of scenarios

We first choose three possible numbers of scenarios, namely 10, 30, and 50, and run the policies with each number. We display the average percentage gaps and the running time for each scenario number summarized by δ, β, and time scale c. In Table 3.6[1], each row corresponds to an average calculated over a subset of instances with the fixed values of δ, β, and c parameters on all instances, respectively. We observe that for both VFA-CF and VFA-2S, the gaps are similar across the three settings, and there is a slight advantage shown for a higher number of scenarios; the running time increases with the number of scenarios, which is reasonable as the more scenarios, the more runs in VFA-CF and the bigger model size in VFA-2S. To balance policy quality and running time, we set $|\Omega|$ equal to 30.

[1] It is worth noting that we identified an outlier in the instance when $\delta = 1, \beta = 0.5$, and $c = 1.2$, which has a significantly longer running time equal to 619.88s for VFA-2S with 30 scenarios. To provide a more generalized analysis, we removed this specific running time and obtained the following results: $\delta =$ 1-27.85, $\beta =$ 0.5-25.65, and $c =$ 1.2-27.16, with an average of 27.80, as shown in the table. The average results before the removal were $\delta =$ 1-34.02, $\beta =$ 0.5-31.84, and $c =$ 1.2-35.40, with an overall average of 33.04.

Table 3.6: Average percentage gap $\Gamma_{p,|\Omega|}$ from the best policy found for $|\Omega| \in \{10, 30, 50\}$ and $p \in \{\text{VFA-CF}, \text{VFA-2S}\}$.

	VFA-CF						VFA-2S					
$\|\Omega\|$	10		30		50		10		30		50	
	gap[%]	t[s]	gap[%]	t[s]	gap[%]	t[s]	gap[%]	t[s]	gap[%]	t[s]	gap[%]	t[s]
δ=0	0.37	36.21	0.03	45.69	0.06	53.07	0.51	26.40	0.45	27.70	0.17	29.80
δ=0.5	0.24	35.59	0.40	45.91	0.33	55.75	0.54	26.54	0.87	27.85	0.42	29.78
δ=1	0.43	45.93	0.13	52.79	0.10	69.22	0.68	27.41	0.40	27.85	0.32	30.06
β=0.5	0.69	47.21	0.15	59.29	0.03	81.70	0.62	24.22	0.52	25.65	0.38	27.41
β=1	0.14	36.35	0.20	47.25	0.31	53.85	0.52	27.48	0.71	28.82	0.31	31.19
β=1.5	0.21	34.16	0.21	37.85	0.16	42.50	0.59	28.65	0.49	28.90	0.23	31.04
c=0.6	0.45	36.94	0.32	44.82	0.32	49.63	0.58	28.73	0.57	29.57	0.22	32.39
c=0.8	0.38	52.39	0.07	62.36	0.11	83.32	0.55	26.87	0.80	28.41	0.51	31.17
c=1	0.53	34.60	0.26	41.51	0.09	46.97	0.88	25.89	0.46	26.04	0.35	27.53
c=1.2	0.04	33.03	0.10	43.82	0.14	57.48	0.30	25.64	0.47	27.16	0.14	28.42
AVG	**0.35**	**39.24**	**0.19**	**48.13**	**0.16**	**59.35**	**0.58**	**26.78**	**0.57**	**27.80**	**0.30**	**29.88**

Table 3.7: Average percentage gap $\Gamma_{p,\rho}$ from the best policy found for $\rho \in \{5, 10, 15, 20\}$ and $p \in \{\text{VFA-CF}, \text{VFA-2S}\}$.

	VFA-CF				VFA-2S			
ρ	5	10	15	20	5	10	15	20
$\delta = 0$	7.11	5.25	2.83	4.49	6.94	5.85	3.17	4.50
$\delta = 0.5$	6.72	4.85	4.23	5.58	5.90	4.70	4.31	6.12
$\delta = 1$	5.18	5.74	3.61	4.00	5.05	5.13	3.49	4.19
$\beta = 0.5$	10.00	8.48	5.20	8.09	9.87	8.41	5.70	8.91
$\beta = 1$	6.37	4.79	3.62	4.09	5.90	4.94	3.47	4.11
$\beta = 1.5$	2.64	2.57	1.84	1.89	2.13	2.33	1.81	1.80
$c = 0.6$	9.12	7.36	6.32	8.37	8.63	7.32	6.18	9.64
$c = 0.8$	8.52	6.55	3.57	5.04	8.05	6.21	4.36	5.05
$c = 1$	3.85	4.70	2.78	3.38	3.89	3.82	2.50	2.83
$c = 1.2$	3.86	2.50	1.54	1.96	3.29	3.54	1.60	2.23
AVG	**6.34**	**5.28**	**3.55**	**4.69**	**5.97**	**5.23**	**3.66**	**4.94**

Table 3.8: Average percentage gap $\Gamma_{p,\gamma}$ from the best policy found for $\gamma \in \{0.7, 0.8, 0.9, 1.0\}$ and $p \in \{\text{VFA-CF}, \text{VFA-2S}\}$.

	VFA-CF				VFA-2S			
γ	0.7	0.8	0.9	1.0	0.7	0.8	0.9	1.0
δ=0	1.27	2.00	1.99	3.24	2.05	1.98	2.33	4.04
δ=0.5	1.77	1.53	2.27	4.64	1.81	1.29	2.38	3.72
δ=1	2.19	1.97	2.20	3.16	2.20	2.37	2.19	3.13
β=0.5	2.77	2.70	2.81	2.67	3.37	2.72	3.18	2.93
β=1	1.58	1.56	1.79	3.83	1.73	1.87	1.79	3.92
β=1.5	0.88	1.24	1.86	4.55	0.96	1.04	1.92	4.04
c=0.6	2.10	2.26	2.42	3.39	2.25	1.84	2.92	3.20
c=0.8	1.78	2.05	2.73	4.98	2.72	2.60	2.79	4.79
c=1	1.56	1.69	1.89	3.48	1.41	1.67	1.80	3.96
c=1.2	1.53	1.33	1.57	2.88	1.69	1.41	1.68	2.56
AVG	**1.74**	**1.83**	**2.15**	**3.68**	**2.02**	**1.88**	**2.30**	**3.63**

3.6.2.2 Batch size

In this section, we present the results of the experiments we made to set a proper value of ρ, i.e., the maximum number of packages transported in future routes used in the batch approach. We test four values: $\rho \in \{5, 10, 15, 20\}$.

Table 3.7 shows the percentage gaps for the VFA-CF and VFA-2S approaches, respectively.

For both VFA-CF and VFA-2S, we see that, on average, the performance improves when increasing ρ from 5

Table 3.9: Average percentage gap $\Gamma_{p,\lambda}$ from the best policy found for $\lambda \in \{0.4, 0.5, 0.6\}$ and p is VFA-CF.

λ	$\vartheta = 0$	$\vartheta = 0.5$	$\vartheta = 1$	$\beta = 0.5$	$\beta = 1$	$\beta = 1.5$	$c = 0.6$	$c = 0.8$	$c = 1$	$c = 1.2$	**AVG**
0.4	0.32	0.05	0.58	0.66	0.18	0.11	0.21	0.33	0.66	0.06	**0.32**
0.5	0.03	0.23	0.58	0.24	0.13	0.47	0.41	0.09	0.43	0.19	**0.28**
0.6	0.49	0.74	0.48	0.42	0.42	0.87	1.25	0.38	0.41	0.23	**0.57**

to 15 but deteriorates when it increases to 20. The best results are obtained when ρ equals 15. Specifically, when $\rho = 15$, the gaps are the smallest in most cases, while $\rho = 5$ and 20 show some exceptionally good results in a few cases, for example, when $\beta = 1.5$ in VFA-2S. Based on the overall results, we opt for a value of $\rho = 15$ as it generally provides the best results.

3.6.2.3 Discount factor

We now analyze the impact of the discount factor on the performance. In Table 3.8, we see that the results are similar among the three values of $\gamma \in \{0.7, 0.8, 0.9, 1.0\}$ for both policies, which means that the choice on γ does not significantly influence the performance. To be consistent, we choose the value of $\gamma = 0.8$ for both VFA-CF and VFA-2S.

3.6.2.4 Performance of the VFA-CF on the basis of λ

We test the performance of VFA-CF based on the percentage of scenarios in which a location must appear to be included in route 0, represented by λ in Algorithm 2 in the chapter. Similarly to other subsections of Section 3.6.2, we report the average gap (%) compared to the best policy obtained for VFA-CF over the three tested values of λ. In Table 3.9, we see the gaps are all within 2%, which means that the choice on the λ does not significantly influence the performance. We opt for $\lambda = 0.5$ as this value provides the smallest gap.

To summarize, we opt for $|\Omega| = 30$, ρ to 15, $\gamma = 0.8$ for both policies, and $\lambda = 0.5$ for VFA-CF.

3.6.3 Benchmark approaches

In the following, we present three benchmark approaches used for comparison with VFA-CF/2S.

3.6.3.1 Two myopic approaches (no future information)

In contrast to the two proposed VFA approaches that make use of sampling, future information, and exact solutions of ILPs, we propose two myopic approaches as benchmarks that use neither sampling nor future information. One relies on the solution of an ILP while the other is based on a heuristic approach to build the route to compare with. Myopic approaches are commonly used as benchmarks in papers mentioned in the former sections. Examples of such benchmarks include approaches that do not employ sampling or delay, as shown in Voccia et al. (2019), as well as the explicitly labeled "myopic approach" discussed in Archetti et al. (2020a) and Ulmer et al. (2019a). Besides, the two myopic methods we propose mimic real-world cases where companies do not use sophisticated optimization tools. The main idea of both approaches is to dispatch the vehicle immediately in every decision epoch by serving all known requests (if any) that can be feasibly served. The difference between the two lies in how the vehicle route (and eventually the subset of known requests to serve) is determined in each decision epoch.

The first approach denoted as *ME* in the following, works as follows: every time the vehicle is back to the depot, in case there is at least one parcel available, the vehicle is immediately dispatched. All parcels available are served if the deadline is not violated. In case the deadline is violated, there is a need to select a subset of parcels to deliver. This is done through exactly solving an orienteering problem (via a branch-and-cut method) to maximize the number of parcels delivered while satisfying the limited route duration.

The second myopic approach, denoted as *MH*, differs from the first in the way the vehicle route is constructed, which follows the nearest neighbor idea. Specifically, every time the vehicle is at the depot, and there is at least one

parcel available, the dispatching decision is made by selecting first the closest customer location to the depot, then the nearest customer to the one just selected, and so on until the remaining time is not enough to serve any new customer or all parcels available are included on the route.

3.6.3.2 An approach considering point estimation of future information

The comparison with the myopic approaches described in the former section aims to show the benefit of incorporating information about future events into the decision policy, as made in VFA-CF and VFA-2S. However, VFA-CF/VFA-2S also incorporate the use of sampling along with the approximation of future routes (through the batch approach). Thus, we propose a further comparison to show the benefit of these two components. Specifically, we compare VFA-CF and VFA-2S with an ILP-based approach considering point estimation (PE) of future information and no approximation. The approach, to which we refer as OPrd_PE, is based on the OP-rd formulation (3.2a), which uses expected release dates as point estimations and performs exact evaluations of future rewards. At each decision epoch, we set the release dates of each unknown request to their expected values, and then we solve the OP-rd (3.2a) where we impose that route 0 cannot start before the time of the current decision epoch. We then take the first route in the obtained solution. If all requests served in it are available, we execute the route immediately. Otherwise, we wait until the requests arrive according to the actual release dates, and execute the route. In case the returning time of the route is after the deadline, which may happen because the actual release dates can be later than the corresponding expected value, the route is not executed. Note that this approach embeds the optimal solution of an ILP to determine the action associated with each decision epoch, similar to VFA-2S and VFA-CF. However, it calculates the exact value associated with future routes (contrary to what is done in VFA-2S and VFA-CF, where this value is approximated due to the batch approach), which clearly might require a longer computing time.

3.6.3.3 Summary of the Characteristics of the Approaches

Table 3.10 summarizes the key characteristics of the different approaches compared in this section. The myopic methods MH and ME do not employ sampling nor consider future information. In addition, at each decision epoch, the decision in ME is based on the solution of ILPs, while MH is based on the nearest neighbor heuristic. As for OPrd_PE, it does not utilize a sampling technique; instead, it relies on point estimation and exact evaluation of future rewards. Notably, the decision in OPrd_PE is based on the solution of ILPs. In contrast, the two VFA approaches, VFA-CF and VFA-2S, employ sampling and embed future information within ILPs. Besides, they leverage the power of approximation through the batch approach to speed up decision-making.

By comparing VFA approaches to MH, ME and OPrd_PE, we can demonstrate the benefits of embedding future information and approximation within ILPs for better decision-making.

Table 3.10: Summary of major characteristics of the tested approaches

	MH	**ME**	**OPrd_PE**	**VFA-CF**	**VFA-2S**
Sampling	✗	✗	✗	✓	✓
Future Information	✗	✗	✓(PE)	✓	✓
Future Approximation	✗	✗	✗	✓	✓
ILP solution	✗	✓	✓	✓	✓

3.6.4 Computational results

In the following subsections, we first show the benefit of partial characterization of optimal policy, then we present the performance of the VFA approaches by comparing them with the three benchmark approaches, then we show the comparison with a deterministic and static approach and experiments for two vehicles.

3.6.4.1 Benefit of partial characterization of optimal policy

We illustrate the benefit of partial characterization (PC) of optimal policy (see Section 3.4.2) by comparing the performance of VFA-CF and VFA-2S with and without PC. Results are shown in Table 3.11 where instances are classified according to the value on δ first, then β, and finally c. For each of the two approaches, the table reports the average computing time (in seconds) with and without PC, the average and maximum percentage improvement in policy value of the approach with PC with respect to the corresponding one without PC (over the set of improved instances), and the number of times the approach with PC improved on the policy without PC.

In Table 3.11, we observe that when using PC, the running time decreases in VFA-CF. The reduction of time in VFA-CF is because the use of PC avoids some runs of the deterministic ILPs. On the other hand, in VFA-2S, the use of PC slightly increases the running time. Furthermore, we observe that policies are improved more significantly when release dates are spread at a moderate level ($\beta = 1$) and when the time horizon is large ($c = 0.8, 1, 1.2$). Overall, we have an improvement of 3.44% for VFA-CF and 3.05% for VFA-2S over the set of instances improved, with maximum improvements being 11.76% and 6.98%, respectively. Considering both the improvement in policy quality for the subset of improved instances and the reduction in running time, we see that the inclusion of PC of the optimal policy demonstrates an overall improvement in the performance of our proposed methods.

Table 3.11: Comparison of VFA-CF and VFA-2S with and without partial characterization of optimal policy

		VFA-CF					**VFA-2S**				
		t[s]		Improvement			t[s]		Improvement		
	#instances	noPC	withPC	AVG(%)	MAX(%)	#impr.	noPC	withPC	AVG(%)	MAX(%)	#impr.
$\delta = 0$	144	82.80	43.70	3.10	5.00	9	34.46	32.08	2.81	5.00	11
$\delta = 0.5$	144	49.61	45.34	3.93	9.68	7	34.81	36.20	3.74	6.98	7
$\delta = 1$	144	69.36	60.45	4.67	11.76	9	39.03	42.85	4.07	6.82	8
$\beta = 0.5$	144	101.51	64.90	0.00	0.00	0	45.87	45.50	0.00	0.00	0
$\beta = 1$	144	57.44	47.54	4.20	11.76	20	31.94	33.02	3.69	6.98	21
$\beta = 1.5$	144	42.82	37.04	2.70	4.44	5	30.49	32.61	2.44	2.86	5
$c = 0.6$	108	47.91	44.88	2.50	2.50	1	30.80	33.04	2.82	2.86	2
$c = 0.8$	108	53.98	47.34	5.63	11.76	5	30.53	33.18	3.41	5.56	5
$c = 1$	108	107.56	57.73	5.11	6.82	7	44.28	40.50	4.74	6.82	7
$c = 1.2$	108	59.57	49.36	2.59	5.13	12	38.78	41.45	2.81	6.98	12
AVG/MAX	**129.6**	**67.26**	**49.83**	**3.44**	**11.76**	**7.5**	**36.10**	**37.04**	**3.05**	**6.98**	**7.8**

3.6.4.2 Performance of VFA-CF and VFA-2S

In this section, we compare the results of VFA-CF and VFA-2S (with PC) against ME, MH, and OPrd_PE. The goal is to show the benefit of incorporating the approximation of future routes and future information into exact solution methods to improve sequential decision making. The Key Performance Indicator (KPI) used for the comparison is the gap from the best policy found over all approaches. In addition, we also report, for each of the five approaches, the gap with respect to the upper bound with perfect information obtained by solving Formulation (3.2). This latter measure is related to the competitive analysis where the focus is on measuring the discrepancy in policy quality due to lack of information.

Results are provided in Table 3.12, condensed by values of β and δ, and graphical representations are presented in Figure 3.2 (displaying values over β) and Figure 3.3. In addition, summarized results over the value of c can be found in Table 3.19 and Figure 3.4. Detailed results for all values of β, δ and c can be found in Tables 3.20 and 3.21, while Figures 3.5-3.7 show an example of solutions obtained through the different approaches. The running times reported below include all the preprocessing steps, such as reading input data, implementing the MDP structure, and the runs of Algorithm 1. As running Algorithm 1 usually takes less than one second, we do not report it separately.

For each approach, we report the average gap with respect to the best policy found over all approaches (denoted as $gap\%$), the gap with respect to the upper bound with perfect information ($gap_{ub}\%$), the computational time in seconds ($t[s]$) and the number of times the best policy was found, including ties ($freq$). As for the $gap\%$, we calculated it as follows. Given an input instance, let N_p denote the value of the policy found by approach p. The gap with respect to the best policy found over all approaches is calculated as $1 - \frac{N_p}{\max_{m \in \{VFA-CF, VFA-2S, ME, MH, OPrd_PE\}} N_m}$. Instead, the gap with respect to the upper bound is calculated as $1 - \frac{N_p}{UB}$, where UB is the value of the upper bound with perfect information.

Table 3.12: Average performance of the five approaches (50 customers)

			VFA-CF				VFA-2S			
β	δ	#instances	gap[%]	gap_ub[%]	t[s]	freq	gap[%]	gap_ub[%]	t[s]	freq
0.5	0	48	10.00	24.24	41.03	25	9.45	23.72	29.93	25
	0.5	48	10.93	23.41	48.07	20	12.28	24.51	42.48	14
	1	48	12.84	25.45	105.61	15	12.67	25.24	64.10	13
	AVG	**48**	**11.26**	**24.36**	**64.90**	**20.0**	**11.47**	**24.49**	**45.50**	**17.3**
1	0	48	3.92	19.36	54.81	21	3.39	18.85	34.64	26
	0.5	48	4.71	21.95	48.13	24	3.98	21.33	33.02	22
	1	48	3.58	15.60	39.69	20	4.28	16.11	31.40	17
	AVG	**48**	**4.07**	**18.97**	**47.54**	**21.7**	**3.88**	**18.76**	**33.02**	**21.7**
1.5	0	48	1.07	7.75	35.26	36	0.64	7.40	31.68	40
	0.5	48	3.09	13.83	39.82	28	2.28	13.25	33.11	26
	1	48	2.18	10.73	36.04	29	2.76	11.15	33.04	26
	AVG	**48**	**2.11**	**10.77**	**37.04**	**31.0**	**1.89**	**10.60**	**32.61**	**30.7**
AVG		**48**	**5.81**	**18.03**	**49.83**	**24.2**	**5.75**	**17.95**	**37.04**	**23.2**

			MH				ME				OPrd_PE			
β	δ	#instances	gap[%]	gap_ub[%]	t[s]	freq	gap[%]	gap_ub[%]	t[s]	freq	gap[%]	gap_ub[%]	t[s]	freq
0.5	0	48	36.20	45.97	0.19	0	23.78	34.88	82.27	6	15.38	26.73	154.42	30
	0.5	48	30.21	38.81	0.21	2	20.60	30.45	30.81	10	16.67	26.41	171.74	33
	1	48	34.79	43.24	0.20	0	24.58	34.09	67.25	0	4.31	16.44	199.98	34
	AVG	**48**	**33.73**	**42.67**	**0.20**	**0.7**	**22.99**	**33.14**	**60.11**	**5.3**	**12.12**	**23.19**	**175.38**	**32.3**
1	0	48	17.98	30.10	0.22	0	9.20	23.11	119.77	13	11.68	25.88	623.72	21
	0.5	48	19.22	32.65	0.23	3	11.45	26.90	86.28	7	19.82	32.15	508.52	24
	1	48	15.67	25.11	0.24	6	8.30	19.26	73.11	19	4.36	15.81	557.29	33
	AVG	**48**	**17.62**	**29.29**	**0.23**	**3.0**	**9.65**	**23.09**	**93.05**	**13.0**	**11.95**	**24.61**	**563.18**	**26.0**
1.5	0	48	7.27	12.79	0.21	18	2.20	8.60	38.54	28	13.74	17.44	788.30	32
	0.5	48	8.32	18.28	0.24	12	4.07	14.71	8.46	29	10.16	19.18	636.08	31
	1	48	7.12	14.72	0.23	16	3.09	11.42	38.71	29	3.62	11.60	698.63	35
	AVG	**48**	**7.57**	**15.26**	**0.23**	**15.3**	**3.12**	**11.58**	**28.57**	**28.7**	**9.17**	**16.07**	**707.67**	**32.7**
AVG		**48**	**19.64**	**29.07**	**0.22**	**6.3**	**11.92**	**22.60**	**60.58**	**15.7**	**11.08**	**21.29**	**482.08**	**30.3**

Besides, we summarize the detailed statistics of the 432 instances used over different values of β and δ in Tables 3.13-3.18. Table 3.13 lists the average lengths of the instances' planning horizons in minutes. For each solution approach, Table 3.14 presents the average numbers of routes executed; Table 3.15 shows the average numbers of known orders that can be delivered at each decision epoch over the instances, which reflect the size of the ILP models; Table 3.16 shows the average numbers of decision epochs and the average running time per decision epoch over the instances; Table 3.17 and Table 3.18 display the average vehicle traveling and waiting time over the instances in minutes, where the waiting time is calculated by subtracting traveling time from the corresponding length of the planning horizon.

Comparison between VFA-CF and VFA-2S Focusing first on the KPIs related to the best-known policy

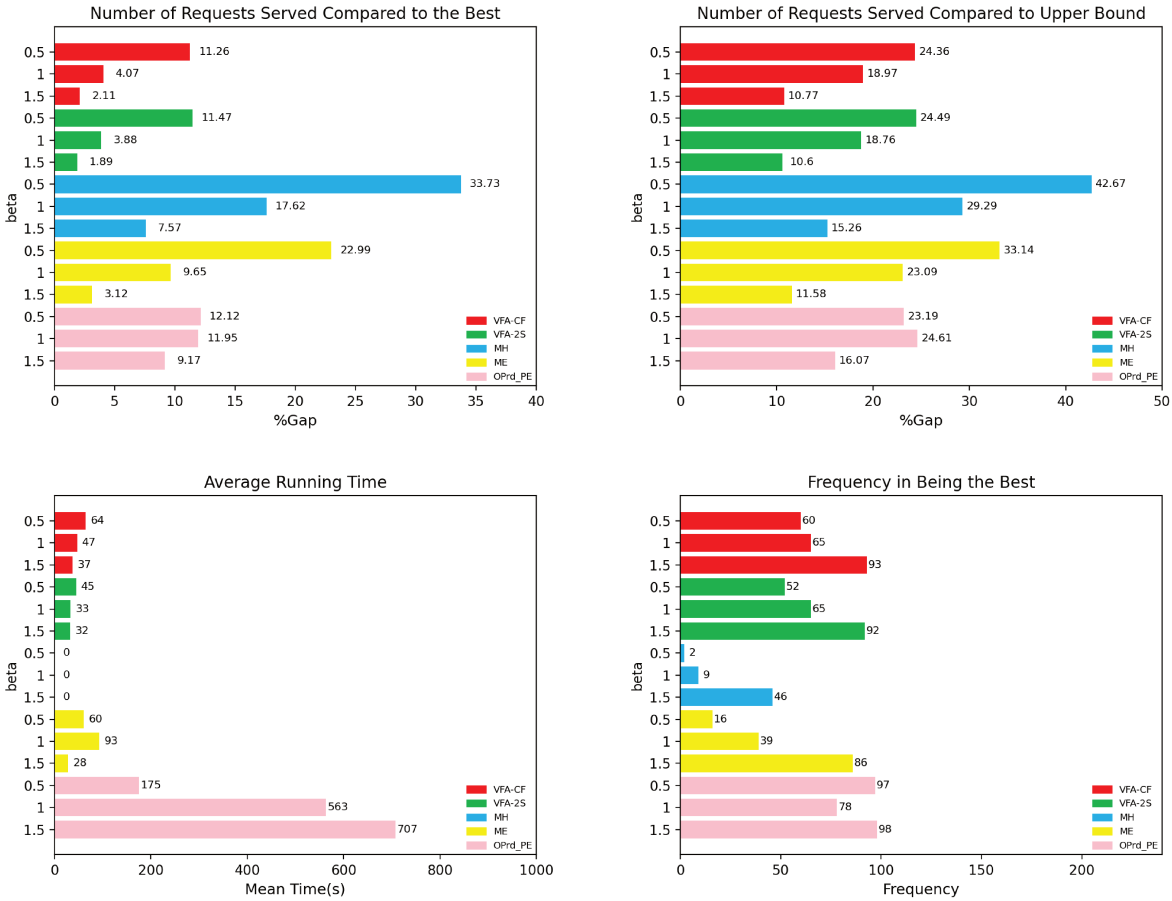


Figure 3.2: Performance of the five approaches based on the values of β (dispersion of release dates)

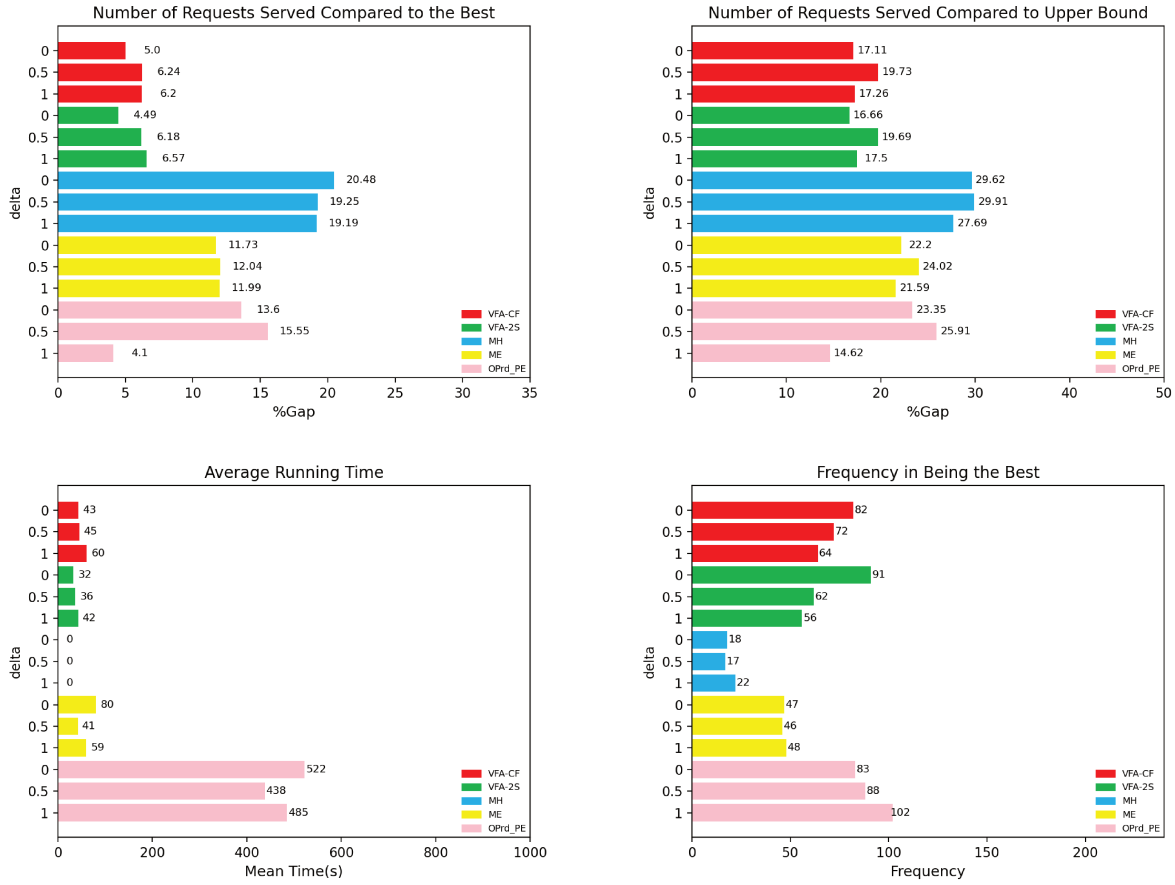


Figure 3.3: Performance of the five approaches based on the values of ∂ (rate of customers with dynamically updated release dates)

Table 3.13: Length of time horizons.

β	δ	timeHorizon
0.5	0	277.94
	0.5	276.63
	1	308.04
	AVG	**287.53**
1	0	550.63
	0.5	535.08
	1	614.94
	AVG	**566.88**
1.5	0	865.63
	0.5	771.88
	1	805.90
	AVG	**814.47**
AVG		**556.29**

Table 3.14: Number of routes.

β	δ	VFA-CF	VFA-2S	MH	ME	OPrd_PE
0.5	0	2.38	2.38	2.38	2.31	2.27
	0.5	2.44	2.42	2.35	2.38	2.54
	1	2.58	2.58	2.44	2.42	2.85
	AVG	**2.47**	**2.46**	**2.39**	**2.37**	**2.56**
1	0	4.44	4.38	3.38	3.38	6.06
	0.5	4.13	4.04	3.21	3.40	5.19
	1	4.23	4.29	3.69	3.98	6.46
	AVG	**4.26**	**4.24**	**3.42**	**3.58**	**5.90**
1.5	0	5.54	5.60	4.81	5.19	9.56
	0.5	5.48	5.50	4.56	4.96	7.58
	1	5.56	5.65	4.83	5.23	8.48
	AVG	**5.53**	**5.58**	**4.74**	**5.13**	**8.54**
AVG		**4.09**	**4.09**	**3.52**	**3.69**	**5.67**

Table 3.15: Number of known requests at each decision epoch.

β	δ	VFA-CF	VFA-2S	MH	ME	OPrd_PE
0.5	0	15.27	15.11	17.02	17.05	19.88
	0.5	14.99	15.08	16.06	16.88	19.31
	1	14.31	14.43	16.35	15.41	18.86
	AVG	**14.86**	**14.87**	**16.48**	**16.45**	**19.35**
1	0	13.10	13.19	13.44	12.50	16.62
	0.5	12.49	12.58	13.02	12.70	15.54
	1	10.91	11.49	12.63	11.66	16.55
	AVG	**12.17**	**12.42**	**13.03**	**12.29**	**16.24**
1.5	0	8.13	9.71	9.05	8.52	14.79
	0.5	9.53	10.16	9.48	8.79	14.77
	1	8.23	8.91	9.29	8.86	13.37
	AVG	**8.63**	**9.59**	**9.27**	**8.72**	**14.31**
AVG		**11.88**	**12.30**	**12.93**	**12.48**	**16.63**

Table 3.16: Number of decision epochs and running time per decision epoch.

		VFA-CF		VFA-2S		MH		ME		OPrd_PE	
β	δ	#epochs	t[s] per epoch	#epochs	t[s] per epoch	#epochs	t[s] per epoch	#epochs	t[s] per epoch	#epochs	t[s] per epoch
0.5	0	6.19	6.63	5.94	5.04	3.19	0.06	3.13	26.33	3.02	51.12
	0.5	6.56	7.32	6.67	6.37	3.00	0.07	3.21	9.60	3.33	51.52
	1	7.46	14.16	7.44	8.62	3.31	0.06	3.15	21.38	3.42	58.53
	AVG	**6.74**	**9.37**	**6.68**	**6.68**	**3.17**	**0.06**	**3.16**	**19.10**	**3.26**	**53.72**
1	0	6.85	8.00	6.83	5.07	4.00	0.05	4.25	28.18	6.83	91.28
	0.5	8.10	5.94	7.79	4.24	3.98	0.06	4.25	20.30	6.00	84.75
	1	7.63	5.21	7.04	4.46	4.54	0.05	4.85	15.06	7.06	78.91
	AVG	**7.53**	**6.38**	**7.22**	**4.59**	**4.17**	**0.05**	**4.45**	**21.18**	**6.63**	**84.98**
1.5	0	8.98	3.93	6.46	4.90	5.94	0.03	6.25	6.17	10.10	78.02
	0.5	9.31	4.28	7.73	4.28	5.48	0.04	5.96	1.42	8.13	78.29
	1	8.56	4.21	7.48	4.42	5.58	0.04	6.13	6.32	9.13	76.56
	AVG	**8.95**	**4.14**	**7.22**	**4.54**	**5.67**	**0.04**	**6.11**	**4.63**	**9.12**	**77.62**
AVG		**7.74**	**6.63**	**7.04**	**5.27**	**4.34**	**0.05**	**4.57**	**14.97**	**6.34**	**72.11**

(gap[%] and freq) and on the comparison between VFA-2S and VFA-CF in Table 3.12, we notice that, on average, the results obtained with VFA-2S are associated with the smallest gaps, being anyway very similar to the ones obtained with VFA-CF. When considering the number of best policies found, we see VFA-CF behaves better than VFA-2S. However, differences are negligible. We conduct a t-test on the objective values obtained by the two approaches and get a p-value of 0.98, which is significantly greater than 0.05. Thus there is not enough evidence to suggest a significant difference in objective values achieved by the two algorithms, or the chances of the two having different results are low. We instead observe a more remarkable difference in terms of computing time, with VFA-CF being slower than VFA-2S. Specifically, the critical case is the one with $\beta = 0.5$ and $\delta = 1$, where the computing time of VFA-CF is almost two times the one of VFA-2S. This might stem from the fact that VFA-CF Formulation (3.6) is solved for each scenario while VFA-2S involves solving the deterministic equivalent problem

Table 3.17: Traveling time.

β	δ	VFA-CF	VFA-2S	MH	ME	OPrd_PE
0.5	0	263.27	264.00	272.06	271.56	209.65
	0.5	262.73	261.15	273.90	271.17	201.06
	1	290.46	290.69	300.98	303.83	256.08
	AVG	**272.15**	**271.94**	**282.31**	**282.19**	**222.26**
1	0	529.44	529.15	546.13	531.50	451.10
	0.5	507.04	507.79	525.81	523.21	415.58
	1	576.29	576.50	593.54	586.85	535.19
	AVG	**537.59**	**537.81**	**555.16**	**547.19**	**467.29**
1.5	0	789.79	791.33	799.94	797.44	756.19
	0.5	717.54	716.46	738.00	725.27	640.31
	1	755.19	754.42	778.21	762.83	719.31
	AVG	**754.17**	**754.07**	**772.05**	**761.85**	**705.27**
AVG		**521.31**	**521.28**	**536.51**	**530.41**	**464.94**

Table 3.18: Waiting time.

β	δ	VFA-CF	VFA-2S	MH	ME	OPrd_PE
0.5	0	14.67	13.94	5.88	6.38	68.29
	0.5	13.90	15.48	2.73	5.46	75.56
	1	17.58	17.35	7.06	4.21	51.96
	AVG	**15.38**	**15.59**	**5.22**	**5.35**	**65.27**
1	0	21.19	21.48	4.50	19.13	99.52
	0.5	28.04	27.29	9.27	11.88	119.50
	1	38.65	38.44	21.40	28.08	79.75
	AVG	**29.29**	**29.07**	**11.72**	**19.69**	**99.59**
1.5	0	75.83	74.29	65.69	68.19	109.44
	0.5	54.33	55.42	33.88	46.60	131.56
	1	50.71	51.48	27.69	43.06	86.58
	AVG	**60.29**	**60.40**	**42.42**	**52.62**	**109.19**
AVG		**34.99**	**35.02**	**19.79**	**25.89**	**91.35**

(DEP). Solving DEP can be computationally faster than solving multiple scenario problems on a low number of scenarios and considering I/O overheads. Moving to the competitive analysis, we see that VFA-CF and VFA-2S produce results between 7% and 25% from the upper bound with perfect information.

Comparison with the benchmark approaches Focusing now on myopic approaches, they are largely outperformed in terms of policy quality by VFA approaches. In terms of computing time, while the one of MH is negligible, for ME, it is comparable to the one of VFA-2S and VFA-CF (despite achieving much worse policies). From Tables 3.14 and 3.17, we find that MH and ME execute a smaller number of routes, but these routes are longer in duration. As they end up serving fewer requests, we can infer that they potentially overlook the benefits of incorporating future requests to optimize and consolidate deliveries. This observation underscores the importance of considering future information when making sequential decisions.

Regarding OPrd_PE, while it often produces best policies, its performance is inconsistent across instances, and the average running time is significantly longer compared to other approaches. For the effect on δ, we see that when δ is 0, VFA approaches have the smallest gaps. When δ is 1, OPrd_PE improves significantly, but this is not true for others. This can be attributed to the decreasing uncertainties as the vehicle approaches the depot, leading to more precise estimations of release dates (see Section 3.6.1 for details). When δ equals 1, indicating dynamic updates of all customers' release dates, the uncertainties associated with the release dates of unknown parcels decrease as they approach the depot, so point estimations of the release dates of parcels become more precise. OPrd_PE builds

the first route by taking some known and soon-to-arrive unknown requests. Proximity of estimations to actual release dates ensures more on-time scheduling, resulting in a high number of served requests. Conversely, without dynamic updates of release dates, the estimations are less precise, so the routes proposed by OPrd_PE have a higher chance of being delayed. Notably, OPrd_PE's performance is not stable across settings. In contrast, VFA-CF and VFA-2S exhibit more robust performance as they generally outperform all others in gaps, time, and frequency of achieving the best. From Tables 3.14 and 3.17, we observe that OPrd_PE executes more but shorter routes than others, which may not be cost-effective. Additionally, OPrd_PE has a long runtime (8 minutes on average), which is attributed to its higher model complexity that includes both known and unknown requests. This shows the benefit of the batch approach proposed in this chapter: it reduces ILP complexity without affecting policy quality.

In addition, we observe that, in most approaches, gap_{ub}[%] decreases when β increases. This might be due to the fact that, when the release dates are more dispersed, the best policy is to dispatch the vehicle immediately with a subset of known requests, thus avoiding waiting too long at the depot. Instead, when the release dates are less dispersed, determining whether to wait or to start dispatching some parcels becomes more challenging. This is also witnessed by the improved performance, for larger β values, of myopic approaches, which are based on the idea of dispatching the vehicle immediately with the largest possible subset of known requests.

Performance of the Approaches on Time Scale In this section, we compare the effect of the different values of the deadline T_E in Table 3.19 and Figure 3.4. Remember that T_E is determined as the latest release date in the instance ($T_{standard}$) multiplied by parameter c.

In Table 3.19, we see that, on average, the performance of the five approaches shows an improving trend as the value of T_E increases. For each T_E, we observe that VFA-CF gets the smallest gaps when c is 0.8 and 1.2, and for VFA-2S c is 0.6 and 1. Meanwhile, we see that when $T_E = 1.2 \cdot T_{standard}$, the values of *freq* are the biggest for most approaches, reflecting the fact that these instances are easier to solve. Finally, we observe that though *MH* and *ME* provide results in a short time, the policy quality is sacrificed. OPrd_PE provides the best results among

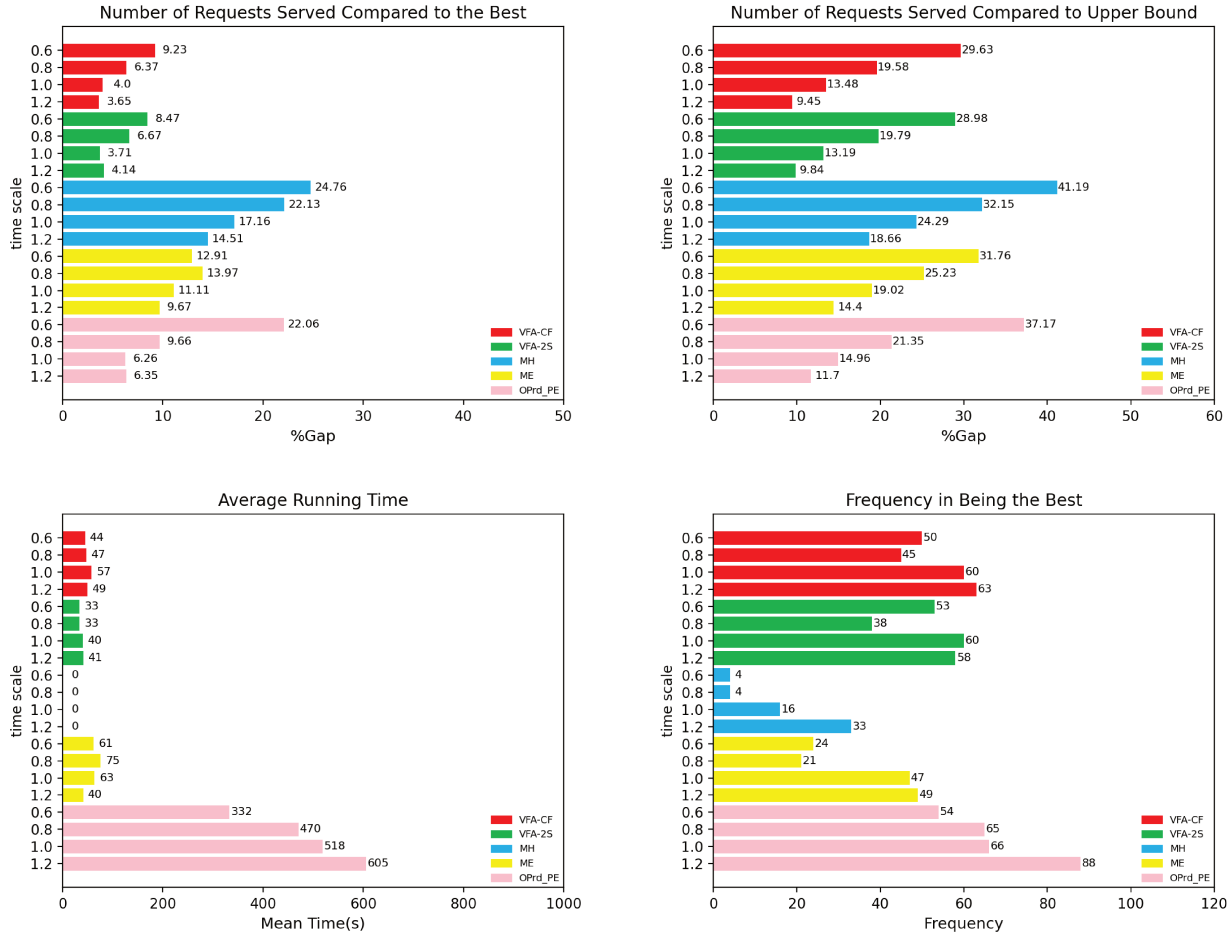


Figure 3.4: Performance of the five approaches based on the value of c

benchmark approaches, especially on the frequency of being the best. However, the average gaps are bigger than VFA-CF and VFA-2S, and the running time is high.

Detailed Computational Results

In Tables 3.20 and 3.21, we display the average performance of the five approaches across various combinations of β, δ, and c. Notably, when β is equal to 1.5, δ is equal to 0, and c is equal to 1.2, all approaches get optimal policies,

Table 3.19: The average performance of the five approaches on the basis of c

		VFA-CF				VFA-2S			
c	**#instances**	gap[%]	gap_ub[%]	t[s]	freq	gap[%]	gap_ub[%]	t[s]	freq
0.6	108	9.23	29.63	44.88	50	8.47	28.98	33.04	53
0.8	108	6.37	19.58	47.34	45	6.67	19.79	33.18	38
1	108	4.00	13.48	57.73	60	3.71	13.19	40.50	60
1.2	108	3.65	9.45	49.36	63	4.14	9.84	41.45	58
AVG	**108**	**5.81**	**18.03**	**49.83**	**54.5**	**5.75**	**17.95**	**37.04**	**52.3**

		MH				ME				OPrd_PE			
c	**#instances**	gap[%]	gap_ub[%]	t[s]	freq	gap[%]	gap_ub[%]	t[s]	freq	gap[%]	gap_ub[%]	t[s]	freq
0.6	108	24.76	41.19	0.25	4	12.91	31.76	61.55	24	22.06	37.17	332.87	54
0.8	108	22.13	32.15	0.21	4	13.97	25.23	75.94	21	9.66	21.35	470.93	65
1	108	17.16	24.29	0.21	16	11.11	19.02	63.89	47	6.26	14.96	518.83	66
1.2	108	14.51	18.66	0.21	33	9.67	14.40	40.93	49	6.35	11.70	605.68	88
AVG	**108**	**19.64**	**29.07**	**0.22**	**14.3**	**11.92**	**22.60**	**60.58**	**35.3**	**11.08**	**21.29**	**482.08**	**68.3**

which means this setting is easy to solve. On the other hand, with $\beta = 0.5$, $\delta = 1$, and $c = 0.6$, VFA approaches get the largest gaps, while OPrd_PE performs well. In general, both VFA and myopic approaches perform better as β increases, and all approaches face challenges in obtaining good policies with smaller values of c in most cases. Additionally, it is noteworthy that VFA approaches exhibit more stable performance in gaps than other approaches. Moreover, the running time of VFA approaches and MH remains stable across settings, whereas ME and OPrd_PE show more diverse runtime.

Illustrative Example

An example is now shown to illustrate the differences among the solutions provided by the different policies. We take an instance with $\beta = 1.5$, $\delta = 0$, and $c = 0.6$. We plot the routes returned by each policy. The routes are colored in order: red - first route, black - second route, green - third route, blue - fourth route, magenta - fifth route, and orange - sixth route. The number next to each node is its release date (RD).

The points are plotted according to their coordinates. The left subfigure in Figure 3.5 refers to VFA-CF and shows five routes starting consecutively with a waiting time of 17 before starting the last route. In the right subfigure that refers to VFA-2S in Figure 3.5, the black route illustrates that VFA-2S chooses to serve a customer with an RD value equal to 28, at the top left, instead of another customer with an RD value of 39, in the center bottom,

Table 3.20: Average performance of the five approaches - part 1 (50 customers)

				VFA-CF				VFA-2S			
β	δ	c	#instances	gap[%]	gap_ub[%]	t[s]	freq	gap[%]	gap_ub[%]	t[s]	freq
	0	0.6	12	6.67	20.24	38.71	9	4.44	18.16	23.49	9
	0	0.8	12	13.90	25.96	49.19	6	13.9	25.96	22.92	6
0.5	0	1	12	9.18	26.45	36.79	6	9.18	26.45	27.33	6
	0	1.2	12	10.25	24.30	39.43	4	10.29	24.32	46.00	4
		AVG	12	10.00	24.24	41.03	6.3	9.45	23.72	29.93	6.3
	0.5	0.6	12	22.25	28.45	26.72	4	22.25	28.45	20.31	4
	0.5	0.8	12	8.81	22.39	40.90	5	10.12	23.46	28.74	4
0.5	0.5	1	12	4.13	21.97	46.71	8	5.78	23.21	76.95	5
	0.5	1.2	12	8.53	20.83	77.94	3	10.96	22.91	43.92	1
		AVG	12	10.93	23.41	48.07	5.0	12.28	24.51	42.48	3.5
	1	0.6	12	23.69	31.70	32.92	3	22.86	30.87	25.88	3
	1	0.8	12	11.15	25.16	49.66	4	10.79	24.76	30.50	4
0.5	1	1	12	9.19	24.43	215.04	3	7.92	23.43	84.19	3
	1	1.2	12	7.32	20.50	124.82	5	9.12	21.89	115.82	3
		AVG	12	12.84	25.45	105.61	3.8	12.67	25.24	64.10	3.3
	AVG		12	11.26	24.36	64.90	5.0	11.47	24.49	45.50	4.3
	0	0.6	12	6.57	37.98	62.46	6	6.27	37.78	38.64	7
	0	0.8	12	3.83	23.47	84.24	6	4.14	23.69	46.44	6
1	0	1	12	2.71	11.98	37.72	6	0.58	9.94	25.83	10
	0	1.2	12	2.56	4.00	34.81	3	2.56	4.00	27.64	3
		AVG	12	3.92	19.36	54.81	5.3	3.39	18.85	34.64	6.5
	0.5	0.6	12	8.56	35.41	53.22	5	5.82	33.40	34.38	6
	0.5	0.8	12	2.53	27.44	50.81	8	2.90	27.62	35.36	6
1	0.5	1	12	4.90	16.30	50.31	5	4.09	15.46	34.92	5
	0.5	1.2	12	2.82	8.67	38.20	6	3.12	8.83	27.42	5
		AVG	12	4.71	21.95	48.13	6.0	3.98	21.33	33.02	5.5
	1	0.6	12	3.52	29.42	50.48	7	5.28	30.66	38.86	5
	1	0.8	12	6.30	18.52	39.82	2	7.59	19.50	32.23	1
1	1	1	12	3.80	10.52	35.97	3	3.54	10.35	27.84	3
	1	1.2	12	0.71	3.92	32.51	8	0.71	3.92	26.68	8
		AVG	12	3.58	15.60	39.69	5.0	4.28	16.11	31.40	4.3
	AVG		12	4.07	18.97	47.54	5.4	3.88	18.76	33.02	5.4
	0	0.6	12	2.04	25.78	46.31	6	0.72	24.74	40.8	10
	0	0.8	12	2.24	5.20	32.45	6	1.86	4.85	30.91	6
1.5	0	1	12	0	0	30.46	12	0	0	27.80	12
	0	1.2	12	0	0	31.82	12	0	0	27.20	12
		AVG	12	1.07	7.75	35.26	9.0	0.64	7.40	31.68	10.0
	0.5	0.6	12	5.58	30.53	51.72	4	3.72	29.22	39.2	5
	0.5	0.8	12	5.31	17.18	41.22	5	4.07	16.33	34.39	2
1.5	0.5	1	12	0.98	5.60	33.55	8	0.98	5.60	29.46	8
	0.5	1.2	12	0.50	2.00	32.80	11	0.33	1.83	29.39	11
		AVG	12	3.09	13.83	39.82	7.0	2.28	13.25	33.11	6.5
	1	0.6	12	4.20	27.11	41.36	6	4.88	27.55	35.84	4
	1	0.8	12	3.23	10.89	37.81	3	4.68	11.98	37.14	3
1.5	1	1	12	1.12	4.08	33.06	9	1.31	4.25	30.23	8
	1	1.2	12	0.17	0.83	31.91	11	0.17	0.83	28.95	11
		AVG	12	2.18	10.73	36.04	7.3	2.76	11.15	33.04	6.5
	AVG		12	2.11	10.77	37.04	7.8	1.89	10.60	32.61	7.7
AVG			12	5.81	18.03	49.83	6.1	5.75	17.95	37.04	5.8

as seen in VFA-CF. Then VFA-2S starts the green route by serving customers located in the bottom left rather than those in the top left. The following two routes are similar between VFA-2S and VFA-CF. The left and right subfigures in Figure 3.6 referring to MH and ME respectively, show that the requests served by the two myopic

Table 3.21: Average performance of the five approaches - part 2 (50 customers)

				MH				ME				OPrd_PE			
β	ϑ	c	#instances	gap[%]	gap_ub[%]	t[s]	freq	gap[%]	gap_ub[%]	t[s]	freq	gap[%]	gap_ub[%]	t[s]	freq
	0	0.6	12	27.28	38.05	0.20	0	7.27	20.10	1.07	6	39.12	44.29	135.65	6
	0	0.8	12	39.74	47.43	0.19	0	29.78	38.00	21.16	0	9.64	19.50	110.64	6
0.5	0	1	12	39.29	50.59	0.19	0	30.35	42.85	151.00	0	8.33	24.72	144.39	9
	0	1.2	12	38.48	47.80	0.18	0	27.71	38.59	155.86	0	4.41	18.41	227.02	9
		AVG	12	36.20	45.97	0.19	0	23.78	34.88	82.27	1.5	15.38	26.73	154.42	7.5
	0.5	0.6	12	23.83	29.04	0.23	2	12.33	18.5	26.05	5	6.19	11.52	97.49	10
	0.5	0.8	12	28.48	37.73	0.19	0	19.95	30.67	33.83	3	17.22	29.28	186.41	8
0.5	0.5	1	12	32.34	44.21	0.21	0	23.67	36.83	26.18	2	25.79	37.36	156.92	7
	0.5	1.2	12	36.18	44.25	0.22	0	26.47	35.80	37.18	0	17.47	27.47	246.14	8
		AVG	12	30.21	38.81	0.21	0.5	20.60	30.45	30.81	2.50	16.67	26.41	171.74	8.3
	1	0.6	12	34.97	39.99	0.22	0	23.15	28.42	55.71	0	5.06	11.85	133.92	10
	1	0.8	12	35.31	44.29	0.18	0	26.58	36.83	90.07	0	4.31	18.09	151.79	7
0.5	1	1	12	34.41	45.18	0.20	0	24.98	37.23	57.36	0	2.88	18.37	194.83	8
	1	1.2	12	34.49	43.48	0.20	0	23.59	33.90	65.85	0	5.01	17.47	319.38	9
		AVG	12	34.79	43.24	0.20	0	24.58	34.09	67.25	0	4.31	16.44	199.98	8.5
	AVG		12	33.73	42.67	0.20	0.2	22.99	33.14	60.11	1.3	12.12	23.19	175.38	8.1
	0	0.6	12	30.44	54.26	0.29	0	18.86	47.01	52.34	1	14.61	41.61	387.09	5
	0	0.8	12	24.86	39.84	0.21	0	13.10	30.33	271.39	0	2.82	22.53	514.36	6
1	0	1	12	12.54	20.80	0.18	0	3.82	12.59	154.01	6	6.76	15.89	646.12	4
	0	1.2	12	4.07	5.50	0.19	0	1.01	2.50	1.35	6	22.52	23.50	947.33	6
		AVG	12	17.98	30.10	0.22	0	9.20	23.11	119.77	3.3	11.68	25.88	623.72	5.3
	0.5	0.6	12	29.44	49.99	0.23	0	21.16	44.73	12.65	0	33.31	49.80	316.72	5
	0.5	0.8	12	24.19	42.85	0.23	0	12.38	34.16	175.12	2	37.02	53.02	409.02	2
1	0.5	1	12	15.14	24.59	0.24	2	7.85	18.69	103.67	3	2.61	13.96	620.31	6
	0.5	1.2	12	8.13	13.17	0.20	1	4.41	10.00	53.67	2	6.34	11.83	688.02	11
		AVG	12	19.22	32.65	0.23	0.8	11.45	26.90	86.28	1.8	19.82	32.15	508.52	6
	1	0.6	12	26.56	46.19	0.29	0	12.38	36.06	75.96	2	14.28	37.13	331.45	2
	1	0.8	12	18.74	28.9	0.24	0	11.65	22.74	87.59	3	1.35	14.32	489.73	9
1	1	1	12	11.67	17.07	0.22	1	6.43	12.63	76.41	6	0.78	7.69	638.89	11
	1	1.2	12	5.71	8.28	0.20	5	2.74	5.60	52.49	8	1.02	4.09	769.08	11
		AVG	12	15.67	25.11	0.24	1.5	8.30	19.26	73.11	4.8	4.36	15.81	557.29	8.3
	AVG		12	17.62	29.29	0.23	0.8	9.65	23.09	93.05	3.3	11.95	24.61	563.18	6.5
	0	0.6	12	19.91	39.26	0.21	0	6.70	29.22	152.78	1	50.04	61.71	608.24	3
	0	0.8	12	7.63	10.37	0.20	3	1.58	4.68	0.45	6	0.52	3.63	958.10	9
1.5	0	1	12	1.53	1.53	0.21	3	0.51	0.51	0.44	9	4.42	4.42	743.77	8
	0	1.2	12	0	0	0.21	12	0	0	0.48	12	0	0	843.10	12
		AVG	12	7.27	12.79	0.21	4.5	2.20	8.60	38.54	7	13.74	17.44	788.30	8
	0.5	0.6	12	15.43	38.71	0.25	2	8.04	33.11	25.6	5	25.88	45.78	494.29	6
	0.5	0.8	12	10.99	21.90	0.26	0	5.04	16.77	2.25	4	13.13	23.15	633.82	7
1.5	0.5	1	12	4.35	8.66	0.22	3	2.25	6.62	5.09	10	1.61	6.29	766.38	6
	0.5	1.2	12	2.51	3.83	0.23	7	0.96	2.33	0.88	10	0	1.50	649.83	12
		AVG	12	8.32	18.28	0.24	3	4.07	14.71	8.46	7.3	10.16	19.18	636.08	7.8
	1	0.6	12	15.01	35.23	0.29	0	6.32	28.73	151.81	4	10.08	30.82	490.97	7
	1	0.8	12	9.29	16.04	0.21	1	5.69	12.88	1.58	3	0.93	8.64	784.49	11
1.5	1	1	12	3.17	5.95	0.19	7	0.17	3.23	0.80	11	3.12	5.95	757.84	7
	1	1.2	12	1.02	1.67	0.23	8	0.17	0.83	0.62	11	0.34	1.00	761.20	10
		AVG	12	7.12	14.72	0.23	4	3.09	11.42	38.71	7.3	3.62	11.60	698.63	8.8
	AVG		12	7.57	15.26	0.23	3.8	3.12	11.58	28.57	7.2	9.17	16.07	707.67	8.2
AVG			12	19.64	29.07	0.22	1.6	11.92	22.60	60.58	3.9	11.08	21.29	482.08	7.6

policies are widely dispersed (note the green route in MH visits from the node with RD valued 169). This can be attributed to the fact that these policies prioritize serving as many available requests as possible without considering future consolidation opportunities. In contrast, the other policies tend to concentrate the served requests

within a smaller geographical region in each route. This example serves to explain the longer average traveling time observed for the MH and ME policies, as illustrated in Table 3.17. Additionally, we note that traversing the same areas multiple times is inevitable when dealing with uncertain release dates. Furthermore, in Figure 3.7 depicting the results of OPrd_PE, we observe that despite using more routes compared to other approaches, it does not yield superior performance compared to VFA methods.

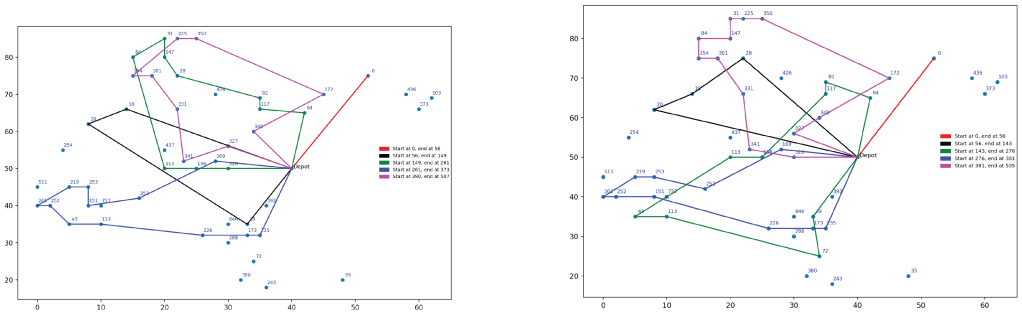


Figure 3.5: Left: VFA-CF returns five routes serving 34 requests, starting at time 0, 56, 149, 261, and 390. Right: VFA-2S returns five routes serving 36 requests, starting at time 0, 56, 143, 276, and 381.

Scalability test We also conduct additional scalability tests on 36 instances containing 100 customers and present the results in Table 3.22. Focusing first on VFA approaches, we observe that their gaps are similar. While the running time increases for both approaches, the one of VFA-CF is much longer compared to VFA-2S. In addition to the explanation provided for the test involving 50 customers, we identify two potential reasons contributing to the extended running time of VFA-CF: (1) As described in Section 3.5.2.1, whenever the trip duration for serving the selected customers from the consensus function exceeds the remaining time, an orienteering problem model is solved to determine the final requests to be served. In contrast, VFA-2S does not involve such a step. (2) The time limit for running VFA-CF Formulation (3.6) is set at 5 minutes, and we run that formulation 30 times (as there are 30 scenarios) at each decision epoch. Conversely, VFA-2S has a time limit of 10 minutes and is solved only once.

Next, in comparison with benchmark approaches, VFA approaches consistently exhibit minimum gaps, approx-

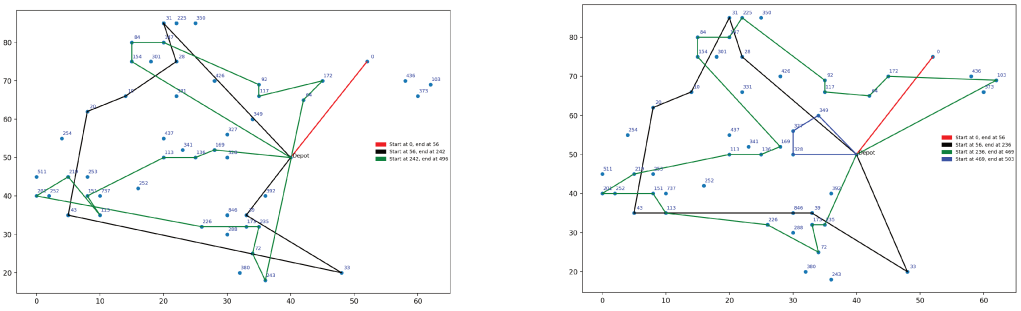


Figure 3.6: Left: MH returns three routes serving 27 requests, starting at time 0, 56, and 242. Right: ME returns four routes serving 31 requests, starting at time 0, 56, 236, and 469.

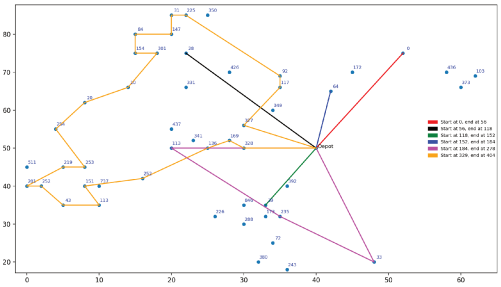


Figure 3.7: OPrd_PE returns six routes serving 31 requests, starting at time 0, 56, 118, 152, 184, and 329

Table 3.22: Average performance of the five approaches (100 customers)

			VFA-CF			VFA-2S			MH			ME			OPrd_PE		
β	δ	**#instances**	gap[%]	t[s]	freq	gap[%]	t[s]	freq	gap[%]	t[s]	freq	gap[%]	t[s]	freq	gap[%]	t[s]	freq
0.5	0	4	1.29	117.77	3	1.29	52.17	3	32.81	0.35	0	23.03	170.86	0	33.30	1552.86	1
	0.5	4	1.23	515.22	3	1.23	63.55	3	41.97	0.34	0	26.15	176.88	0	23.40	1394.72	1
	1	4	10.16	649.99	3	10.16	67.29	3	38.96	0.33	0	31.17	66.39	0	14.38	6243.51	1
	AVG	**4**	**4.23**	**427.66**	**3.0**	**4.23**	**61.00**	**3.0**	**37.91**	**0.34**	**0.0**	**26.78**	**138.04**	**0.0**	**23.69**	**3063.70**	**1.0**
1	0	4	3.56	203.42	2	3.60	63.50	2	31.52	0.33	0	27.57	151.25	1	64.35	12160.15	0
	0.5	4	3.07	327.95	3	0.00	62.23	4	37.55	0.34	0	37.47	0.73	0	26.10	14439.80	0
	1	4	0.00	1649.04	4	0.00	209.57	4	41.34	0.36	0	40.56	0.75	0	31.18	14870.07	0
	AVG	**4**	**2.21**	**726.80**	**3.0**	**1.20**	**111.76**	**3.3**	**36.81**	**0.34**	**0.0**	**35.20**	**50.91**	**0.3**	**40.54**	**13823.34**	**0.0**
1.5	0	4	0.00	336.78	4	1.75	167.87	2	43.42	0.35	0	42.69	0.78	0	27.47	16658.04	0
	0.5	4	0.45	99.89	3	0.00	47.46	4	33.73	0.37	0	29.42	0.80	0	46.07	9948.25	0
	1	4	0.00	102.53	4	0.00	55.34	4	42.68	0.37	0	42.32	0.69	0	51.63	14281.81	0
	AVG	**4**	**0.15**	**179.73**	**3.7**	**0.58**	**90.22**	**3.3**	**39.95**	**0.36**	**0.0**	**38.14**	**0.75**	**0.0**	**41.72**	**13629.37**	**0.0**
AVG		**4**	**2.20**	**444.73**	**3.2**	**2.00**	**87.66**	**3.2**	**38.22**	**0.35**	**0.0**	**33.38**	**63.24**	**0.1**	**35.32**	**10172.14**	**0.3**

imately 2%, which is much smaller than those of the other approaches (over 30%). While the myopic approaches are still fast, especially the simple MH approach, they achieve much worse policies compared to VFA approaches in most cases. Scaling from 50 to 100 customers, the performance of OPrd_PE greatly deteriorates and the running time increases significantly. This outcome is anticipated, given the explosion in model size and search space, making it a challenge to find an optimal solution within the constrained running time.

Summary Overall, both VFA-2S and VFA-CF largely outperform the other approaches, among them, the myopic ones being considered as proxies of practical policies. The difference in policy quality between VFA-2S and VFA-CF is marginal. However, as VFA-2S is faster, it is a good compromise between policy quality and computational burden. To summarize, we showed the benefits of:

- approximating future routes through a batch approach that helped largely reduce the computing time (with respect to OPrd_PE) without affecting policy quality;
- taking into account future information in the decision process by comparing the VFA-CF/VFA-2S with two myopic approaches, which provide vastly worse results.

3.6.4.3 Comparison with a Deterministic and Static Approach

To show the importance of considering both stochastic and dynamic release dates, we compare VFA approaches with an alternative approach, called OPrd_Sto, that uses only the stochastic information available at time zero. Similarly to the OPrd_PE, OPrd_Sto is based on the OP-rd formulation, using expected release dates at time 0 as point estimations. It performs exact evaluations of future rewards, but with a distinction: the soluton obtained at time 0 is kept and never modified afterwards. Specifically, for each route of this deterministic solution, we wait until all its scheduled requests have arrived, before sending out the vehicle. The route is executed only in case it does not exceed the deadline. We test OPrd_Sto on the same 432 test instances, each containing 50 customers, as for the VFA approaches. For each instance, we set a time limit of 30 minutes. In Table 3.23, we summarize the results over δ, β, and c. The percentage gap (Gap%) is calculated as follows: for each instance, the gap with VFA-CF is calculated by 100 minus the ratio of requests served by OPrd_Sto to those served by VFA-CF, multiplied by 100. The average percentage gap over all instances is then computed for each parameter (δ, β, or c). The same calculation is used for the comparison with VFA-2S.

Compared to VFA approaches, OPrd_Sto needs 30 times more computing time, on average. This increase in computational effort is due to the exact route calculations. Despite that, the results reveal that OPrd_Sto serves 28% fewer requests compared to VFA approaches on average, indicating a significant performance gap. OPrd_Sto considers stochastic information only at time 0 and produces a deterministic solution. In contrast, VFA approaches account for both stochastic and dynamic aspects, which emphasizes the value of including both types of information in the decision-making process.

3.6.4.4 Multiple vehicles

In this section, we present experiments where two vehicles are available for multiple dispatches in an area during the day. These tests are aimed at showing how VFA-CF and VFA-2S can be adapted to this setting in terms of

Table 3.23: Comparing VFA approaches with an approach that only considers stochastic release dates

		t[s]			Gap%	
	#instances	**OPrd_Sto**	**VFA-CF**	**VFA-2S**	**VFA-CF**	**VFA-2S**
δ=0	144	1568.19	43.70	32.08	16.08	16.60
δ=0.5	144	1568.16	45.34	36.20	28.41	28.41
δ=1	144	1640.35	60.45	42.85	40.41	40.22
β=0.5	144	1341.71	64.90	45.50	20.32	20.31
β=1	144	1790.45	47.54	33.02	33.46	33.58
β=1.5	144	1644.53	37.04	32.61	31.13	31.34
c=0.6	108	1422.19	44.88	33.04	19.55	20.41
c=0.8	108	1650.30	47.34	33.18	33.83	33.62
c=1	108	1618.64	57.73	40.50	28.56	28.69
c=1.2	108	1677.80	49.36	41.45	31.27	30.92
AVG	**129.6**	**1592.23**	**49.83**	**37.04**	**28.30**	**28.41**

scalability and policy quality. The approaches are modified as follows: at each decision epoch, each vehicle is treated individually and independently of the status of the other vehicle, i.e., without considering the existence of the other vehicle (e.g., without leaving some parcels for the other to deliver), according to the approaches described above. We note that the two vehicles still share the same depot and the same set of customer demands.

This two-vehicle approach offers a first attempt at tackling the multi-vehicle case to investigate the viability of our models, providing a feasible approach. The MDP model (definitions of decision epochs, states, decisions, etc) for the multi-vehicle case may need to be modified. However, we believe that similar building blocks based on the batch approach and ILP models can be adapted for the multiple-vehicle case.

In Table 3.24, we show the average running time with one vehicle and two vehicles under the column named t[s], the average number of requests served with two vehicles, the average and maximum improvement of the policy with two vehicles with respect to the policy with one vehicle only, and the number of times the policy with two vehicles improved over the one with one vehicle only. Compared to the single-vehicle version, the two-vehicle version improves over 106 instances out of 130 on average, serving more than 31% customers over the improved instances. Notably, the number of served requests doubled for certain instances, resulting in 100% improvement. It is worth mentioning that the simulation running time increases in the two-vehicle version, and it nearly doubles for the approach with consensus function (but still very fast), compared to the single-vehicle version. This increase

in running time is expected as the simulation is executed more frequently due to the availability of two vehicles. In addition, we find that the average number of requests served by VFA-CF and VFA-2S are similar, with a slight advantage for VFA-2S.

Table 3.24: Performance of Two Vehicles Compared to Single Vehicle (50 customers)

		VFA-CF						VFA-2S					
		t[s]			Improvement			t[s]			Improvement		
	#instances	1Vehicle	2Vehicles	#served	AVG(%)	MAX(%)	#impr.	1Vehicle	2Vehicles	#served	AVG(%)	MAX(%)	#impr.
δ=0	144	43.70	114.46	41.11	35.85	88.24	111	32.08	46.48	41.19	35.54	88.24	111
δ=0.5	144	45.34	61.57	39.13	33.07	80.00	124	36.20	46.56	39.03	32.40	100.00	125
δ=1	144	60.45	64.72	40.95	29.67	100.00	119	42.85	49.46	41.06	30.61	100.00	119
β=0.5	144	64.90	113.08	31.96	49.50	100.00	143	45.50	43.10	31.84	49.60	100.00	143
β=1	144	47.54	60.56	43.32	24.53	75.00	128	33.02	45.78	43.46	24.63	76.92	129
β=1.5	144	37.04	67.12	45.92	16.78	60.00	83	32.61	53.62	45.98	16.48	55.00	83
c=0.6	108	44.88	63.66	30.13	42.98	100.00	108	33.04	48.41	30.29	42.60	100.00	108
c=0.8	108	47.34	62.46	39.58	32.75	93.75	101	33.18	46.53	39.51	32.84	88.24	101
c=1	108	57.73	64.38	44.21	28.03	87.50	79	40.50	48.54	44.23	27.83	87.50	79
c=1.2	108	49.36	130.50	47.67	21.93	80.95	66	41.45	46.53	47.68	22.72	84.00	67
AVG/MAX	**129.6**	**49.83**	**80.25**	**40.40**	**31.51**	**100.00**	**106.2**	**37.04**	**47.50**	**40.43**	**31.52**	**100.00**	**106.5**

3.7 Conclusion

To the best of our knowledge, this article is the first to study the orienteering problem with stochastic and dynamic release dates. For this challenging sequential decision-making problem, we propose two RL approaches. Both rely on MDP with: discrete sets of scenarios simulating future realizations of release dates in each decision epoch, approximation of future routes, and exact solutions based on ILP models. However, they employ the approximation of future routes in two different ways: VFA-CF handles each scenario independently and uses a consensus function to decide on the action in each decision epoch. VFA-2S considers all scenarios simultaneously and uses a two-stage stochastic ILP model instead. They are compared with two myopic alternatives (no future information used) that we consider as proxies of practical policies, and an alternative ILP-based approach considering PE of future information. Our empirical study confirms the benefits of incorporating approximation of future routes and future information into exact solution methods to improve sequential decision making. Both VFA-CF and VFA-2S significantly outperform myopic approaches, not only in terms of best-obtained gaps with respect to the

perfect-information upper bounds but also in terms of the frequency with which the best-quality policies are obtained. Compared to the one that exploits PE of future information, VFA-CF and VFA-2S are less sensitive to the changes in the dynamics of release dates and provide more stable results along with much lower running time. The difference in policy quality between VFA-CF and VFA-2S is marginal. However, as VFA-2S is faster, we believe it is a good compromise between policy quality and computational burden.

This chapter shows that exact methods provide benefits in solving complex sequential stochastic and dynamic optimization problems, specifically routing problems, which are known to be extremely difficult even in their deterministic counterpart. Also, accounting for future uncertainty as done in VFA-2S/VFA-CF, i.e., by coupling exact methods (Sections 3.5.2-3.5.3) with an approximation of the future routes (Section 3.5.1), brings substantial value compared to the approaches that do not use future information (ME/MH), without losing tractability (which is instead lost in the OPrd_PE). Moreover, the overall methodology does not require extensive training (as many of *Q*-learning approaches do), and relies on a tuning of very few hyper-parameters instead. The current paper provides a solid base for building approaches for other more complicated problem settings, involving, for example, vehicle capacity, time windows, or multiple vehicles. Additionally, the methodology can be potentially better tailored to the situations in which parcels arrive in batches. For the multi-vehicle case: OP-rd and ME can be adapted by integrating multiple vehicles into the formulations, and MH can be modified to assign requests to idle vehicles at each decision epoch. In Section 3.6.4.4, we show how the VFA approaches are used for two vehicles. An interesting research direction could be to design tailored approaches for the multi-vehicle case. Adapting VFA-CF and VFA-2S, combined with the approximation of future routes based on the batch approach, is more involved as the assignment of batches to vehicles should be included in the batch approach. However, the methodology proposed in this chapter represents a solid starting point and given that it proved effective in single-vehicle and two-vehicle cases, it holds promise for the multi-vehicle case.

4

Delivery with Pickup Stations and Autonomous Ground Vehicles

joint work with **Claudia Archetti** and **Geoffrey Chua**

4.1 Introduction

E-commerce grows continuously, driven by consumer demand for convenience and rapid order fulfillment. A key challenge in last-mile logistics is balancing delivery efficiency with operational costs while meeting service-level expectations. Pickup stations have emerged as an effective alternative to conventional home delivery, offering both economic and environmental benefits. These facilities include automated lockers and pickup points, where the former refers to parcel lockers that allow customers to retrieve their orders without requiring staff assistance, and the latter represents staffed collection points that are integrated into local stores and operate during their regular business hours. They enable to bundle customer demands and reduce delivery failures, thus improving the delivery efficiency and reducing the associated delivery cost as well as CO_2 emissions (Rohmer & Gendron, 2020; Ranjbari et al., 2023). Meanwhile, logistics providers are increasingly exploring the use of electrified autonomous vehicles in delivery networks (Joseph, 2025). These vehicles offer various advantages, particularly in mitigating labor shortages and reducing environmental impact. However, their widespread deployment is often hindered by regulatory constraints and infrastructural limitations. Autonomous vehicles currently operating at SAE[1] Levels 3 and 4 are restricted to a defined operational design domain, meaning that they can operate safely only under specific conditions and environments[2]. As a result, current implementations typically limit these vehicles to predefined, well-mapped routes where safety and reliability can be ensured.

In this chapter, we study a promising application that uses autonomous vehicles for direct deliveries from a central depot to pickup stations, called *direct trips*. This setting aligns with current autonomous vehicle capabilities by avoiding complex routing challenges and uncertainties. Alternative delivery options, such as delivery to pickup stations, increase customer flexibility but introduce planning complexities, including the selection of delivery loca-

[1] SAE refers to the Society of Autonomous Engineers, which defines six levels of vehicle automation, from Level 0 (no automation) to Level 5 (full autonomy in all conditions), the latter of which remains a future goal.

[2] https://semiengineering.com/gearing-up-for-level-4-vehicles/

tions and capacity management at pickup stations (when referring to lockers). This chapter addresses a practically relevant problem within this context, proving that the problem and some of its variants are NP-hard while others can be solved in polynomial time. We demonstrate how the problem can be effectively tackled with approaches based on established combinatorial optimization methods with suitable adaptations. While the connection between this problem and classical optimization models—such as the knapsack problem or the budgeted maximum coverage problem—may not be evident, we establish a clear link in this chapter. By uncovering these connections, we leverage both existing and new methodologies to develop effective solution approaches for this practical and pressing problem. From a theoretical perspective, we derive performance bounds and conduct worst-case analyses. On the computational side, we evaluate our proposed approaches through empirical comparisons on realistic instances.

In the literature, many papers considering last-mile delivery with pickup stations can be found (see, for example Zhou et al. (2018); Sitek & Wikarek (2019); Orenstein et al. (2019); Dell'Amico et al. (2023); Galiullina et al. (2024)). For a broader discussion on pickup stations, we refer readers to the survey by Rohmer & Gendron (2020) for an overview of key challenges and decision problems, and Mohri et al. (2024) for insights into the evolving role of alternative delivery points, including pickup stations. Besides, there has been a growing interest in studying autonomous ground vehicles in last-mile delivery, see Alfandari et al. (2022); Reed et al. (2022); Zhao et al. (2024). For a broader overview of autonomous ground vehicle applications in last-mile delivery, we refer readers to a survey by Srinivas et al. (2022) that discusses autonomous delivery robots, including ground vehicles. However, research that integrates autonomous ground vehicles with delivery to pickup stations remains limited. To the best of our knowledge, the only related study is by Ulmer & Streng (2019), who investigate same-day delivery from a central depot to pickup stations using autonomous vehicles. In their model, customers dynamically request deliveries to their preferred stations or nearby alternatives. In contrast, we consider a static and deterministic delivery problem, where all customer orders are known in advance. Our focus is on optimizing deliveries from the depot to pickup stations using autonomous ground vehicles, under various operational constraints such as station capacity and de-

livery deadline. Unlike Ulmer & Streng (2019), who emphasize dynamic decision-making, we develop and analyze several heuristics with performance guarantees across multiple problem variants.

In light of this, this chapter presents the following main contributions:

- We study a delivery problem with pickup stations and autonomous ground vehicles. While prior work (Ulmer & Streng (2019)) considers dynamic and stochastic setting, our formulation captures a complementary and practically relevant case in which all orders are known in advance. We find connections between this problem and classical combinatorial optimization problems and prove that it is NP-hard.
- We analyze multiple problem variants, each reflecting different operational constraints, and develop efficient heuristics tailored to their specific structures.
- Computational experiments on realistic instances show that, for each instance, at least one of the proposed heuristics achieves an optimality gap within 3%. Moreover, our findings highlight that the selection of a solution approach should be guided by the characteristics of the problem instances, as the most effective heuristic varies depending on the instance-specific factors.

The remainder of the chapter is organized as follows. Section 4.2 introduces the delivery optimization problem, presents the mathematical model, and analyzes the computational complexity. Section 4.3 explores various problem variants and their corresponding solution approaches. Section 4.4 presents heuristics for the original problem with capacitated pickup stations and multiple station options per parcel. Section 4.5 presents computational tests. Finally, Section 4.6 concludes the chapter.

4.2 Model

In this section, we formally introduce the delivery optimization problem (DOP), which considers a single autonomous ground vehicle, capacitated pickup stations, a common delivery deadline, and multiple station options per parcel in Section 4.2.1. We then present the mixed integer linear programming (MILP) model in Section 4.2.2 and prove that problem is NP-hard in Section 4.2.3.

4.2.1 Problem Definition

In DOP, a set of orders, denoted as $\mathcal{N}$, should be delivered before a deadline T. Each order is designated to be delivered to a pickup station and can also be delivered to any station within a specified radius ρ. A single autonomous ground vehicle is responsible for delivering parcels to a set of pickup stations in a region, with multiple parcels delivered per trip. Let $\mathcal{P}$ represent the set of pickup stations, and let o denote the depot. For each customer n, $\mathcal{P}_n$ denotes the subset of nearby stations available for pickup by customer n.

We assume that the vehicle conducts direct trips, traveling only between the depot and a pickup station without intermediate stops or routing decisions. The vehicle's capacity is enough to transport all parcels assigned to a station. The capacity of each pickup station i is denoted as c_i. We do not account for additional space becoming available as customers retrieve their parcels, since delivery systems typically reserve locker capacity for different shipping options rather than dynamically adjusting it (Sethuraman et al., 2024). Each parcel occupies one unit of station capacity. Let d_{0j} represent the travel time between the depot 0 and a pickup station $i \in \mathcal{P}$. The objective is to maximize the number of parcels delivered before T.

4.2.2 MILP Formulation

The DOP problem described in the previous section can be formulated as the following MILP model.

The decision variables used are:

- y_{ni}: binary variable equal to 1 if the parcel of customer n is delivered to pickup station i, 0 otherwise;
- z_i: binary variable equal to 1 if pickup station $i \in P$ is visited, 0 otherwise;

The formulation is the following:

$$\max \sum_{i \in \mathcal{P}_n} \sum_{n \in N} y_{ni} \tag{4.1a}$$

$$\sum_{i \in \mathcal{P}_n} y_{ni} \leq 1 \qquad n \in \mathcal{N} \tag{4.1b}$$

$$c_i z_i \geq \sum_{n \in N} y_{ni} \qquad i \in \mathcal{P} \tag{4.1c}$$

$$\sum_{i \in \mathcal{P}} 2 d_{0i} z_i \leq T \tag{4.1d}$$

$$y_{ni}, z_i \in \{0, 1\} \qquad n \in \mathcal{N}, i \in \mathcal{P}. \tag{4.1e}$$

The objective function (4.1a) maximizes the number of customers served. Constraints (4.1b) ensure that each customer is served at most once. Constraints (4.1c) ensure that if a station receives parcels in a given trip, z_i is set to one, and the station's capacity is not exceeded. Constraints (4.1d) impose that trips must be completed before the delivery deadline.

4.2.3 Computational Complexity of DOP

In this section, we prove that DOP is NP-hard by showing that even a simplified variant (SDOP), where each parcel has a single designated pickup station, a common delivery deadline, and no capacity constraints at pickup stations, remains NP-hard.

Theorem 1. *Problem SDOP is NP-hard.*

Proof. We prove that SDOP is NP-hard by reducing it from the 0/1 knapsack problem.

An instance of 0/1 knapsack problem I_u is defined as follows: given a set of items U, each with a weight w_u and value v_u, $u \in U$, a weight limit T, and a value goal K, the goal is to decide whether there exists a subset $U' \subseteq U$ such that $\sum_{u \in U'} w_u \leq T$ and $\sum_{u \in U'} v_u \geq K$.

Given an instance of the 0/1 knapsack problem, we build an instance I_p of SDOP as follows. There is a set of pickup stations $\mathcal{P}$ corresponding to items U. The number of parcels assigned to each station i_m represents the item value v_u. The round-trip travel time $2d_{i0}$ corresponds to the item weight w_u. The goal is to determine whether there exists a subset $\mathcal{P}' \subseteq \mathcal{P}$ such that: $\sum_{i \in \mathcal{P}'} 2d_{i0} \leq T$ and $\sum_{i \in \mathcal{P}'} i_m \geq K$.

We claim that I_u is positive if and only if I_p has a feasible solution: if a solution exists for I_u, we can select the corresponding pickup stations in I_p that satisfy the deadline and parcel constraints; Conversely, if I_p has a feasible solution, we can construct valid selection of items for I_u that meets the knapsack constraints.

Since solving the 0/1 knapsack problem amounts to finding a feasible solution for SDOP, SDOP is NP-hard.

It follows that the more general problem DOP is also NP-hard.

4.3 Complexity and heuristics for problem variants

In the following, we study different problem variants, beginning with the simplest variant in Section 4.3.1, where there is no deadline constraint. We then introduce deadline restrictions in Section 4.3.2, covering station capacity and assignment flexibility: uncapacitated or capacitated pickup stations without multiple station options per parcel in Sections 4.3.2.1; uncapacitated stations with multiple station options per parcel in Section 4.3.2.2.

4.3.1 No Deadline: Capacitated Assignment Problem (CAP)

When there is ample time to travel to all requested pickup stations but station capacity is limited, the challenge shifts from managing time constraints to optimizing the parcels to station assignments. For example, during non-peak periods, such as midweek days or non-holiday seasons when fewer parcels are shipped compared to peak shopping periods (e.g., Black Friday or Christmas), station capacity can still be a constraint because each station (especially when referring to lockers) has a finite number of compartments and some locations may have a higher demand due to customer preferences or location convenience (e.g., near offices or metro stations). We refer to this problem as the Capacitated Assignment Problem (CAP).

In this scenario, the problem reduces to assigning parcels to pickup stations while satisfying capacity constraints, without considering the delivery deadline.

We model the problem with Formulation (4.2), referred to as the max-flow formulation. In this formulation, q_{ni} is a binary variable, equal to 1 if the parcel of customer n is assigned to station i.

$$\max \sum_{n \in N} \sum_{i \in \mathcal{P}_n} q_{ni} \tag{4.2a}$$

$$\sum_{i \in \mathcal{P}_n} q_{ni} \leq 1 \qquad n \in N \tag{4.2b}$$

$$\sum_{n \in N} q_{ni} \leq c_i \qquad i \in \mathcal{P} \tag{4.2c}$$

$$q_{ni} \in \{0, 1\} \qquad n \in \mathcal{N}, i \in \mathcal{P} \tag{4.2d}$$

The objective function (4.2a) maximizes the number of parcels served. Constraints (4.2b) limit each parcel to be served at most once, and (4.2c) ensure the capacity of each pickup station is not exceeded.

Theorem 2. *The problem without a deadline, where stations are capacitated and there are multiple station options per parcel, is solvable in polynomial time.*

Proof. We model the problem CAP as a flow network where parcels are represented as source nodes, pickup stations as sink nodes, and arcs connect parcels to the stations they can be assigned to. Each arc has a capacity of 1, showing that a parcel can only be delivered to one station. Each Station has a capacity corresponding to the maximum number of parcels they can accept. Finding a feasible assignment that respects station capacities and parcel-station compatibilities then reduces to finding a maximum flow in this network. Since the maximum-flow problem is solvable in polynomial time (Goldberg & Tarjan, 1988), CAP is also solvable in polynomial time.

4.3.2 With Deadline

We now consider scenarios in which there may not be enough time to deliver to every pickup station. We begin by analyzing the case where each parcel is assigned to a single designated pickup station in Section 4.3.2.1. We then extend the analysis to variants where parcels may have multiple station options: first with uncapacitated pickup stations in Section 4.3.2.2 and then with capacitated stations in Section 4.4.

4.3.2.1 Single designated station per parcel with uncapacitated or capacitated pickup stations (SUCAP)

This case happens when customers specify a single designated pickup station or when the platform prioritizes customer satisfaction by delivering parcels to their preferred stations. As shown in Theorem 1, the problem remains NP-hard even without capacity constraints, as long as there is a delivery deadline. It is equivalent to the 0/1 knapsack problem, where the maximum number of parcels that can be delivered to each station represents the item value, and the round-trip travel time between the depot and the station corresponds to the item weight. For brevity, we refer to this problem as SUCAP.

When pickup stations are capacitated, the problem is a generalization of the NP-hard problem SDOP (see Theorem 1), and remains equivalent to the 0/1 knapsack problem. Based on this equivalence, we propose a greedy heuristic similar to the one proposed for knapsack, achieving an approximation ratio of $1/2$.

A Greedy Heuristic

We propose a heuristic for selecting pickup stations based on the ratio between the maximum number of parcels they can receive and their round-trip travel time from/to the depot. The process begins by ordering stations in descending order of this ratio. Stations are then selected sequentially, ensuring that adding another station does not exceed the delivery deadline. Next, we repeat the procedure until all parcels are assigned or no further assignments can be made within the allotted time. Finally, the set of selected stations is compared with the station that can accommodate the highest number of parcels, and the option that serves the most parcels is chosen. This heuristic is specified in Algorithm 3. Specifically, steps from lines 2 and 13 ensure that the number of parcels assigned to each station does not exceed its capacity. In particular, line 2 selects an arbitrary subset of parcels with size up to $\min(c_i, |\mathcal{N}_i|)$. These steps are omitted when stations are uncapacitated.

Proposition 1. *If each parcel has a single designated station, a delivery deadline is imposed, and stations are either*

Algorithm 3 Greedy heuristic for SUCAP

Require: Deadline T; set of pickup stations $\mathcal{P}$; travel time from station i to depot 0, d_{i0} and back, d_{0i}; set of parcels $\mathcal{N}$; set of parcels that can be delivered to station i, $\mathcal{N}_i$; remaining time $t_d \leftarrow T$, capacity of station i, c_i;

Ensure: Ordered set of selected pickup stations S;

Initialize $S \leftarrow \emptyset, M \leftarrow 0$
$\mathcal{M}_i \leftarrow subset(\mathcal{N}_i, \min(c_i, |\mathcal{N}_i|)), \forall i \in \mathcal{P}$
while $t_d > 0$ and $M < |\mathcal{N}|$ and $|S| < |\mathcal{P}|$ **do**
    $i' \leftarrow \arg\max_{i \in \mathcal{P} \setminus S} \{\frac{\mathcal{M}_i}{d_{i0}+d_{0i}}\}$
    if $t_d - d_{i'0} - d_{0i'} \geq 0$ **then**
        $S \leftarrow [S, i']$
        $M \leftarrow M + \mathcal{M}_{i'}$
        $t_d \leftarrow t_d - d_{i'0} - d_{0i'}$
    else
        break
    end if
end while
$j \leftarrow \arg\max_{i \in P} \{\min(c_i, |\mathcal{N}_i|)\}$, where $t_d - d_{i'0} - d_{0i'} \geq 0$
if $|\mathcal{N}_j| > M$ **then return** $\{j\}$ **else return** S.
end if

uncapacitated or incapacitated, the approximation ratio of Algorithm 3 is 1/2*, and the bound is tight.*

The proof can be omitted because the problem is equivalent to the 0/1 knapsack problem, and Algorithm 3 follows the same logic as the 1/2 greedy heuristic for the 0/1 knapsack problem (see Theorem 2.5.4 in Kellerer et al. (2011)).

4.3.2.2 Multiple station options per parcel with uncapacitated stations (MUCAP)

In scenarios where multiple station options are available for a parcel, the problem is equivalent to the Budgeted Maximum Coverage problem (BMC, Khuller et al. (1999)), as established in Proposition 2. The BMC problem is defined as follows: Given a collection $U = \{U_1, U_2, ..., U_m\}$ of sets with associated costs $\{c_i\}_{i=1}^m$ defined over

a domain of elements $X = \{x_1, x_2, \ldots, x_n\}$ with associated weights $\{w_i\}_{i=1}^{n}$, the goal is to find a subcollection $U' \subseteq U$, such that the total weight of elements covered by U' is maximized while ensuring that the total cost in U' does not exceed a given budget B. We refer to this problem as MUCAP.

Proposition 2. *If each parcel has multiple station options, a delivery deadline is imposed, and pickup stations are uncapacitated, the problem is equivalent to the BMC with equal weight per element.*

Proof. In our problem, each parcel corresponds to an element with equal weight, and each pickup station corresponds to a set containing multiple elements (i.e., the parcels it can serve). The round-trip travel time between the depot and each station represents the selection cost. The goal is to select a subset of stations such that the cost does not exceed the deadline while maximizing the number of delivered parcels, which is analogous to the goal in the BMC: Find a subset of sets such that the total cost does not exceed a given budget while maximizing the total weight of covered elements.

If a solution exists for the BMC, we can select the corresponding pickup stations in our problem that satisfy the deadline constraint while delivering the corresponding number of parcels. Conversely, if a feasible solution exists for our problem, we can construct a corresponding selection of sets in the BMC formulation that meets the cost and coverage requirements.

Since BMC is known to be NP-hard even when all elements have equal weight and each set has unit cost (Khuller et al., 1999; Cohen & Katzir, 2008), and our problem is equivalent to BMC with equal-weight elements and general cost, it follows that our problem is also NP-hard.

In the following, we propose two heuristics adapted from those proposed for the BMC in Khuller et al. (1999). Notably, the heuristics designed for the case with multiple station options per parcel can also be applied to the scenario with single station designated per parcel discussed in Section 4.3.2.1.

A Greedy Heuristic

We first propose a heuristic that extends the uncapacitated version of Algorithm 3 to handle cases where parcels have multiple station options. The pseudocode is provided in Algorithm 4. A key modification is that, each time after calculating the station i' with the highest ratio between the maximum number of parcels it can receive and its round-trip travel time from/to the depot (line 6), the parcels assigned to i' (denoted as $\mathcal{N}_{i'}$) are removed from the parcel sets of other stations to avoid duplication (lines 11-14). Another major change lies in the stopping criterion: rather than terminating when the next station's delivery completion time exceeds the deadline (as in lines 9 and 10 of Algorithm 3), this heuristic continues evaluating the remaining stations. It selects those that can be visited within the remaining time and updates the set of parcels that can be served by the unselected stations accordingly.

Proposition 3. *If each parcel has multiple station options, a delivery deadline is imposed, and pickup stations are uncapacitated, Algorithm 4 achieves an approximation ratio of* $\frac{1}{2} \cdot (1 - \frac{1}{e})$.

Proof. Algorithm 4 corresponds to the Algorithm 1 proposed in Khuller et al. (1999) for the BMC. The authors show that this algorithm has an approximation ratio of $1/2 \cdot (1 - 1/e)$. Khuller et al. (1999) find this ratio for a general problem where element weights in BMC may vary. The version with equal weight is a direct application of their result and does not alter the performance guarantee. For a detailed proof, we refer to Khuller et al. (1999).

An Improved Heuristic

In this section, we analyze an alternative algorithm proposed for solving the BMC (Khuller et al., 1999), which achieves an improved approximation ratio of $1 - \frac{1}{e}$ under the same problem setting. Let h be a fixed integer, and let $S \subseteq \mathcal{P}$ denote a sub-collection of stations. Define $w(S)$ as the total number of parcels covered by S, and let $t(S)$ represent the total round-trip travel time required to deliver to all stations in S. The formal description is shown in Algorithm 5. Notably, this algorithm can also be applied to solve the scenario with single station designated

Algorithm 4 A greedy heuristic for MUCAP

Require: Deadline T; set of pickup stations $\mathcal{P}$; travel time from station i to depot 0, d_{i0} and back, d_{0i}; set of parcels $\mathcal{N}$; set of parcels that can be delivered to station i, $\mathcal{N}_i$; remaining time $t_d \leftarrow T$;
Ensure: Ordered set of selected pickup stations S;

1: Initialize $S \leftarrow \emptyset, M \leftarrow 0$
2: $M_i = |\mathcal{N}_i|, \forall i \in \mathcal{P}$
3: $\mathcal{N}'_i \leftarrow \mathcal{N}_i, \forall i \in \mathcal{P}$
4: $U \leftarrow \mathcal{P}$
5: **while** $t_d > 0$ and $M < |\mathcal{N}|$ and $|U| > 0$ **do**
6: $\quad i' \leftarrow \arg\max_{i \in U}\{\frac{M_i}{d_{i0}+d_{0i}}\}$
7: $\quad$ **if** $t_d - d_{i'0} - d_{0i'} \geq 0$ **then**
8: $\quad\quad S \leftarrow [S, i']$
9: $\quad\quad M \leftarrow M + M_{i'}$
10: $\quad\quad t_d \leftarrow t_d - d_{i'0} - d_{0i'}$
11: $\quad\quad$ **if** $\mathcal{N}_{i'} \cap \mathcal{N}_i \neq \emptyset, i \in U \setminus i'$ **then**
12: $\quad\quad\quad \mathcal{N}_i \leftarrow \mathcal{N}_i \setminus \{\mathcal{N}_{i'} \cap \mathcal{N}_i\}$
13: $\quad\quad\quad M_i = |\mathcal{N}_i|$
14: $\quad\quad$ **end if**
15: $\quad$ **end if**
16: $\quad U \leftarrow U \setminus i'$
17: **end while**
18: $j \leftarrow \arg\max_{i \in P}\{|\mathcal{N}'_i|\}$, where $T - d_{i'0} - d_{0i'} \geq 0$
19: **if** $|\mathcal{N}'_j| > M$ **then return** $\{j\}$ **else return** S.
20: **end if**

per parcel SUCAP described in Section 4.3.2.1, with the same approximation ratio of $1 - \frac{1}{e}$ for both scenarios in Section 4.3.2.1 and in the current section.

The procedure begins by computing candidate solution H_1, which is defined as the subset of stations, containing fewer than h stations, that maximizes the number of parcels served, subject to the constraint that the total travel time does not exceed the deadline. Next, the algorithm constructs candidate solution H_2 as follows: for each set of h stations S that satisfies the deadline constraint, it evaluates the remaining pickup stations and, if time permits, adds the station with the highest ratio of parcel quantity to round-trip travel time (see lines 6 to 15). If the total number of parcels served by S exceeds that of the current H_2, the solution is updated with S accordingly. Finally,

the algorithm returns the best solution between H_1 and H_2.

Algorithm 5 An improved heuristic for MUCAP

Require: Deadline T; set of pickup stations $\mathcal{P}$; travel time from station i to depot 0, d_{i0} and the reverse, d_{0i}; set of parcels $\mathcal{N}$; set of parcels that can be delivered to station i, $\mathcal{N}_i$; remaining time $t_d \leftarrow T$;

Candidate solution $H_1 \leftarrow \arg\max\{w(S)$, such that $S \subseteq \mathcal{P}$, $|S| < h$, and $t(S) \leq T\}$;
Candidate solution $H_2 \leftarrow \emptyset$
for all $S \subseteq \mathcal{P}$, such that $|S| = h$ and $t(S) \leq T$ **do**
    $U \leftarrow \mathcal{P} \backslash S$
    $Q_i \leftarrow \mathcal{N}_i$
    while $|U| > 0$ **do**
        $i' \leftarrow \arg\max_{i \in U}\{\frac{|Q_i|}{d_{i0}+d_{0i}}\}$
        if $t(S) + d_{i'0} + d_{0i'} \leq T$ **then**
            $S \leftarrow [S, i']$
            if $Q_{i'} \cap Q_i \neq \emptyset, i \in U \setminus i'$ **then**
                $Q_i \leftarrow Q_i \setminus \{Q_{i'} \cap Q_i\}$
            end if
        end if
        $U \leftarrow U \backslash i'$
    end while
    if $w(S) > w(H_2)$ **then** $H_2 \leftarrow S$
    end if
end for
if $w(H_1) > w(H_2)$ **then** H_1 **else return** H_2.
end if

Proposition 4. *For $h \geq 3$, Algorithm 5 achieves an approximation ratio of $1 - \frac{1}{e}$ for MUCAP.*

Proof. Algorithm 5 corresponds to Algorithm 2 proposed in Khuller et al. (1999) for the BMC. The authors show that this algorithm has an approximation ratio of $1 - \frac{1}{e}$. For a detailed proof, we refer to Theorem 4 in Khuller et al. (1999).

4.4 The original problem: multiple station options per parcel and capacitated pickup stations (DOP)

We now return to the original problem setting, where each parcel may have multiple pickup station options, stations are capacitated, and deliveries to stations must be completed within a specified deadline. To handle this added complexity, we develop three heuristics. The first, referred to as *greedySimpleCapacity*, extends Algorithm 3 by incorporating parcel selection after each station is chosen. The second, *greedyHCapacity*, builds on Algorithm 5, by modifying the selection process to respect station capacities. The third, called *Hybrid MILP*, leverages two MILP models to guide the solution process.

GreedySimpleCapacity

To ensure that each parcel is counted only once, this algorithm modifies Algorithm 3 by determining which parcels should be assigned to each station. The complete procedure is described in Algorithm 6. Specifically, parcels are first ordered based on the number of available pickup station options, prioritizing those with fewer alternatives (see lines 8 and 9), as these parcels have a lower likelihood of being assigned later. Once a station i' is selected, it is removed from the station options of the remaining parcels (line 12). Then, the first K_i parcels assigned to i' are removed from the parcel sets of other stations to prevent duplication (lines 13 to 15). As a result, the maximum number of parcels that can be assigned to the remaining stations must also be recalculated to reflect the updated parcel availability (see line 5).

GreedyHCapacity

To guarantee that capacity constraints of pickup stations are satisfied, we modify Algorithm 5 by incorporating the solution of the max-flow formulation (4.2) to determine the maximum number of parcels that can be served. Let $S \subseteq \mathcal{P}$ represent a sub-collection of stations. Let $w(S)$ be the maximum number of parcels that can be covered

Algorithm 6 GreedySimpleCapacity: Greedy heuristic for DOP

Require: Deadline T; set of pickup stations $\mathcal{P}$; travel time from station i to depot 0, d_{i0} and the reverse, d_{0i}; set of parcels $\mathcal{N}$; set of parcels that can be delivered to station i, $\mathcal{N}_i$; remaining time $t_d \leftarrow T$; the set of stations each parcel n can be delivered to $\mathcal{P}_n$;

Ensure: Ordered set of selected pickup stations S;

Initialize $S \leftarrow \emptyset, M \leftarrow \{\}$
$Q_i \leftarrow \mathcal{N}_i, \forall i \in \mathcal{P}$
$U \leftarrow \mathcal{P}$
while $t_d > 0$ and $|M| < |\mathcal{N}|$ and $|U| > 0$ **do**
    $K_i \leftarrow \min(c_i, |Q_i|), \forall i \in U$
    $i' \leftarrow \arg\max_{i \in U}\{\frac{K_i}{d_{i0}+d_{0i}}\}$
    if $t_d - d_{i'0} - d_{0i'} \geq 0$ **then**
        $M_i \leftarrow sort(Q_i, \text{ascending order of } |\mathcal{P}_n|, n \in Q_{i'})$
        $M \leftarrow M \cup M_i[1, K_i]$
        $S \leftarrow [S, i']$
        $t_d \leftarrow t_d - d_{i'0} - d_{0i'}$
        $\mathcal{P}_n \leftarrow \mathcal{P}_n \setminus i'$, where $i' \in \mathcal{P}_n, n \in \mathcal{N} \setminus M$
        if $M_i[1, K_i] \cap Q_i \neq \emptyset, i \in U \setminus i'$ **then**
            $Q_i \leftarrow Q_i \setminus \{M_i[1, K_i] \cap Q_i\}$
        end if
    end if
    $U \leftarrow U \setminus i'$
end while
$j \leftarrow \arg\max_{i \in P}\{\min(c_i, |\mathcal{N}_i'|)\}$, where $t_d - d_{i'0} - d_{0i'} \geq 0$
if $|\mathcal{N}_j'| > |M|$ **then return** $\{j\}$ **else return** S.
end if

by S (considering the union of the parcels covered by S). Let $t(S)$ be the total round-trip travel time required to delivered parcels to stations in S. Let $Z_{mf}(\cdot)$ represent the optimal value obtained by solving the max-flow formulation (4.2).

The procedure is described in Algorithm 7. It begins by computing candidate solution H_1. Specifically, it first enumerates all station subsets with fewer than h stations, filters those that satisfy the deadline constraint, and then solves the max-flow formulation (4.2) for each of the subsets to determine the number of parcels that can be served.

The subset that serves the most parcels is selected as H_1 (line 1). Next, the algorithm constructs candidate solution H_2. For each set of h stations that satisfies the deadline constraint, it computes the number of parcels that can be accommodated at each station using Formulation (4.2) (see line 8). If additional time remains, it selects another station i' with the highest ratio of parcels to round-trip travel time (lines 14 to 16). If adding i' does not violate the deadline (line 17), the algorithm determines which parcels to assign to i', considering station capacity constraints. As in Algorithm 6, it prioritizes parcels with fewer station options (lines 18 and 19). It then updates both the station options for the remaining parcels and parcel sets for the remaining stations in $U \setminus i'$, in cases where they have parcels overlapped with station i' (lines 21-24).

Algorithm 7 GreedyHCapacity: Heuristic for DOP

Require: Deadline T; set of pickup stations $\mathcal{P}$; travel time from station i to depot 0, d_{i0} and the reverse, d_{0i}; set of parcels $\mathcal{N}$; set of parcels that can be delivered to station i, $\mathcal{N}_i$; remaining time $t_d \leftarrow T$; the set of stations each parcel n can be delivered to $\mathcal{P}_n$;

Candidate solution $H_1 \leftarrow \arg\max\{w(S) = Z_{mf}(N_i, c_i, i \in S)$, where $S \subseteq \mathcal{P},\ |S| < h,$ and $t(S) \leq T\}$
Candidate solution $H_2 \leftarrow \emptyset$;
for all $S \subseteq \mathcal{P}$, such that $|S| = h$ and $t(S) \leq T$ **do**
    $U \leftarrow \mathcal{P} \setminus S$
    $Q_i \leftarrow \mathcal{N}_i, i \in \mathcal{P}$
    $\mathcal{P}'_n \leftarrow \mathcal{P}_n, n \in \mathcal{N}$
    $M \leftarrow \{\}$
    $M_i \leftarrow Z_{mf}(\mathcal{N}_i, c_i, i \in S)$
    $M \leftarrow M \cup M_i, i \in S$
    $\mathcal{P}'_n \leftarrow \mathcal{P}'_n \setminus i$, where $i \in \mathcal{P}'_n \cap S, n \in \mathcal{N} \setminus M$
    if $M \cap Q_i \neq \emptyset, i \in U$ **then**
        $Q_i \leftarrow Q_i \setminus \{M \cap Q_i\}$
    end if
    while $|U| > 0$ and $t(S) < T$ **do**
        $K_i \leftarrow \min(c_i, |Q_i|), i \in U$
        $i' \leftarrow \arg\max_{i \in U} \left\{ \frac{K_i}{d_{i0} + d_{0i}} \right\}$
        if $t(S) + d_{i'0} + d_{0i'} \leq T$ **then**
            $M_{i'} \leftarrow \text{sort}(Q_{i'}$, ascending order of $|\mathcal{P}'_n|, n \in Q_{i'})$
            $M \leftarrow M \cup M_{i'}[1, K_{i'}]$
            $S \leftarrow S \cup \{i'\}$
            $\mathcal{P}'_n \leftarrow \mathcal{P}'_n \setminus i'$, where $i' \in \mathcal{P}'_n, n \in \mathcal{N} \setminus M$
            if $M_{i'}[1, K_{i'}] \cap Q_i \neq \emptyset, i \in U \setminus \{i'\}$ **then**
                $Q_i \leftarrow Q_i \setminus \{M_{i'}[1, K_{i'}] \cap Q_i\}$
            end if
        end if
        $U \leftarrow U \setminus \{i'\}$
    end while
    $w(S) = Z_{mf}(\mathcal{N}_i, c_i, i \in S)$
    if $w(S) > w(H_2)$ **then**
        $H_2 \leftarrow S$
    end if
end for
if $w(H_1) > w(H_2)$ **then**
    return H_1
else
    return H_2
end if

Hybrid MILP

Now, we introduce another heuristic that integrates two MILP models: the max-flow formulation (4.2) and the distance-minimization formulation (4.3). We refer to this method as the Hybrid MILP heuristic. The heuristic proceeds as follows:

1. First, it solves the max-flow formulation (4.2) with capacitated pickup stations and no deadline constraint, obtaining the objective value $Z_{mf}(\cdot)$;

2. Second, it solves a travel time minimization problem with a parcel threshold parameter Z', which specifies the minimum number of parcels to be served. The process starts with $Z' = \lambda \cdot Z_{mf}(\cdot)$, where $0 < \lambda \leq 1$ is a tuning parameter. If the resulting objective value exceeds the deadline T, Z' is decreased by $g \cdot Z_{mf}(\cdot)$, where $0 < g < 1$, and the model is solved again. This process repeats until the objective value no longer exceeds T. The model is provided in Formulation (4.3), where a binary variable g_i is introduced; it equals 1 if station $i \in P$ is visited, and 0 otherwise.

$$\min \sum_{i \in \mathcal{P}} (d_{i0} + d_{0i}) g_i \tag{4.3a}$$

$$\sum_{i \in \mathcal{P}_n} q_{ni} \leq 1 \qquad n \in \mathcal{N} \tag{4.3b}$$

$$\sum_{n \in \mathcal{N}} q_{ni} \leq c_i g_i \qquad i \in \mathcal{P} \tag{4.3c}$$

$$\sum_{n \in \mathcal{N}} \sum_{i \in \mathcal{P}_n} q_{ni} \geq Z \tag{4.3d}$$

$$q_{ni}, g_i \in \{0, 1\} \qquad n \in \mathcal{N}, i \in \mathcal{P} \tag{4.3e}$$

The objective function (4.3a) minimizes the total travel time. Constraints (4.3b) ensure that each customer is served at most once. Constraints (4.3c) ensure capacity limits at each pickup station. Constraints (4.3d) guarantee that at least Z' parcels are delivered.

4.5 Numerical experiments

In this section, we present a numerical comparison of the solution approaches by evaluating the three heuristics versus the results of the MILP Formulation (4.1). Our experiments focus on the most challenging setting—referred to as the DOP, which involves multiple station options per parcel, capacitated pickup stations, and a common delivery deadline, as described in Section 4.4. We restrict our attention to this setting since simpler variants either admit polynomial-time solutions or already have established approximation guarantees.

All experiments were performed on an HP Z4 G4 Workstation with an Intel(R) Xeon(R) W-2255 CPU @ 3.70GHz. The algorithms were implemented in Python and solved with Gurobi v10.0.3, using one thread. For solving the MILP Formulation (4.1), we set a one-hour time limit per instance. For the greedyHCapacity heuristic, we set a 1.5-hour time limit per instance, as its runtime increases significantly compared to the other two heuristics (greedySimpleCapacity and Hybrid MILP). For the Hybrid MILP heuristic, we set $\lambda = 0.9$ in $Z' = \lambda \cdot Z_{mf}(\cdot)$ and $g = 0.05$ in $g \cdot Z_{mf}(\cdot)$ as a rule a thumb. The parameter h is determined as the average of the maximum number of stations that can be visited and the number of trips computed based on average round-trip travel times. The calculation details are provided in Section 4.5.2.

The remainder of this section is structured as follows. Section 4.5.1 describes the instance settings. Section 4.5.2 presents the procedure for determining parameter h. Finally, Section 4.5.3 compares the heuristics on the DOP.

4.5.1 Input Instances

We consider two sets of instances: (1) self-generated instances for a large variations of the number of stations and customers, and (2) instances adapted from Ulmer & Streng (2019) for the real-world network of Brauschweig, Germany.

Dataset 1: Self-generated instances

This dataset consists of instances generated from scratch. The parameters are defined as follows: The number of stations (#station) varies from 5 to 30 with increments of 5. The number of customers is set in the range $[5 \cdot \#station, 25 \cdot \#station]$ with increments of 25. The locations of stations and customers are randomly generated within predefined spatial boundaries. Latitude and longitude coordinates are assigned to each station, where both values are independently drawn from a uniform distribution on the interval [0, 90]. Customer locations are similarly generated, with both latitude and longitude values sampled uniformly from [0, 100]. The capacity of each station is randomly assigned from a discrete uniform distribution from [0, 51]. The depot is located in the right center of the graph formed by the stations. Figure 4.1 shows the location plot for the instance with 10 stations and 250 customers. A customer's order can be delivered to any station within the specified radius ρ. We consider three possible radius values: 0, 10, and 20. If the radius is 0 or no stations are located within a customer's specified radius, the customer is assigned to the nearest station. Four different time horizons are considered: 60, 240, 480, and 720. In total, the dataset comprises 1080 instances.

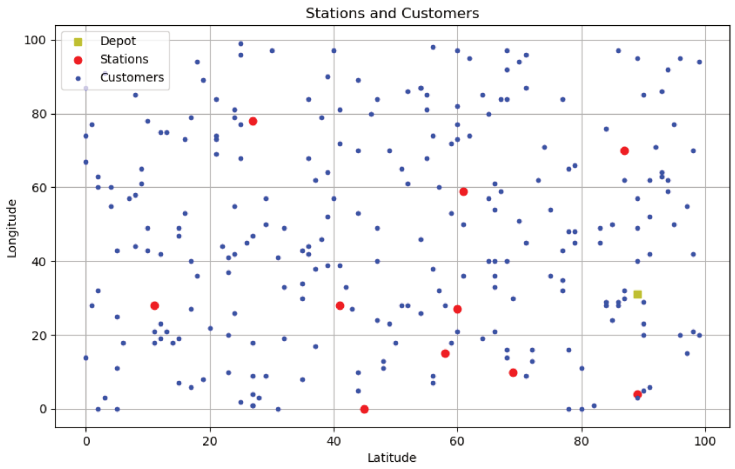


Figure 4.1: Location plot for an instance with 10 stations and 250 customers in Dataset 1.

Dataset 2: Instances adapted from Ulmer & Streng (2019)

This dataset is based on the instances tested in Ulmer & Streng (2019), with modifications to fit the problem settings we study. The parameters are defined as follows: There are 12 stations, each with a capacity of 50. The travel times between stations and depot follow the definitions in Ulmer & Streng (2019) (see Figure 4.2). There are 100 instances where each contains around 600 customer orders. Each order has a preferred station but can be assigned to other stations within a specified radius. We consider three radius values: 0, 3, and 5. If the radius is 0, the customer is assigned to the preferred station. We consider three time horizons: 60, 120, and 180. To evaluate how the number of customers affects algorithm performance, we vary the customer percentage from 20% to 100%. For example, when the percentage is 20%, only the first 20% of customers in the instance are included in the test. In total, this dataset consists of 4500 instances. The travel times between stations and the depot remain identical to those defined in Ulmer & Streng (2019). We present the travel time matrix of Dataset 2 in Figure 4.2.

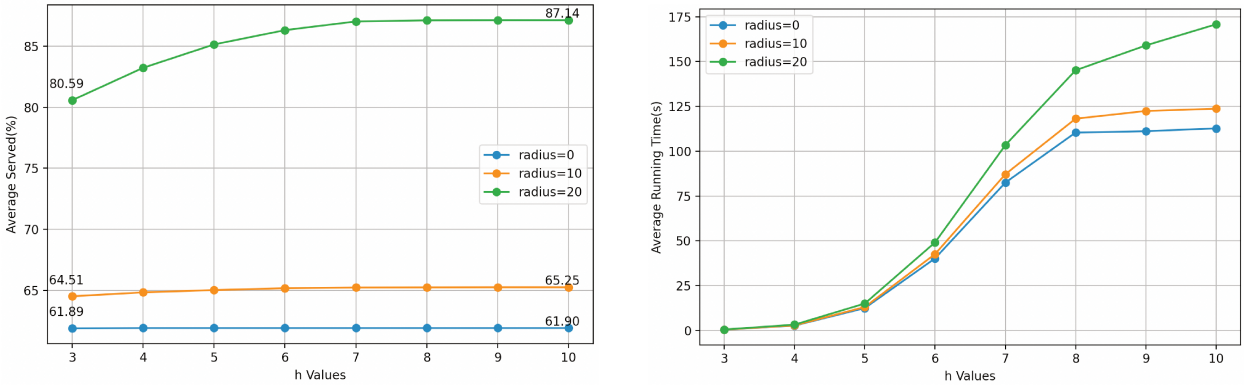
Travel Times from Google Maps between Stations and/or the Depot (Symmetrical).

Start	Pickup Station											
Location	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
Depot	1	7	8	8	13	12	14	13	14	15	14	14
P_1	0	5	4	6	9	12	10	13	13	13	13	12
P_2		0	3	6	8	11	8	11	11	12	12	12
P_3			0	7	5	8	6	9	8	9	12	12
P_4				0	11	14	11	12	10	11	10	10
P_5					0	3	2	5	12	5	8	16
P_6						0	3	4	11	4	7	15
P_7							0	5	12	6	8	16
P_8								0	10	3	6	13
P_9									0	7	7	9
P_{10}										0	3	11
P_{11}											0	8
P_{12}												0

Figure 4.2: Taken from Ulmer & Streng (2019)

4.5.2 Determining the Values of h

In the following, we describe the procedure used to select the parameter h. The tests are conducted on Algorithm 5. To prevent a sharp increase in running time for large values of h, we set its maximum value as the average of the maximum and average number of stations that can be visited within the time limit. As the maximum h values vary across instances, we evaluate 224 instances from Dataset 1 (74 for each radius value), where all tested h values from 3 to 10 are feasible and the time horizon equal to 720.



(a) served(%) vs. h

(b) time(s) vs. h

Figure 4.3: Comparison of average served (%) and running time for different values of h when the time horizon is 720 and radius is 0, 10, and 20.

Figure 4.3, on the left, shows that the average percentage of customers served increases with larger values of h, especially for a radius equal to 20. The possible reason is that the number of customers each station can serve increases when the radius increases and increasing h helps explore more combinations of stations, which in turn results in more customers served on average. However, this comes at a cost with the increasing running time, especially when the radius is equal to 20. The increase of running time has the sharpest growth from $h = 5$ to $h = 8$, followed by steady growth onwards. To balance between running time and solution quality, we set h individually for each instance. Specifically, h is chosen as the average of two quantities: the maximum number of stations that can be visited based on sorting stations by distance to the depot and the average number of stations that can be visited, estimated by using the average round-trip travel time.

4.5.3 Comparisons between Three Heuristics for DOP

In the following, we show computational tests for the original problem, DOP. We also solve the MILP Formulation (4.1) to calculate the optimality gap for each heuristic. Let N_p denote the number of customers served by algorithm

$p \in \{\texttt{greedyHCapacity}, \texttt{greedySimpleCapacity}, \texttt{HybridMILP}\}$. The optimality gap is defined as $1-(N_p/N_{MILP})$, where N_{MILP} is the number of customers served by the MILP Formulation (4.1).

Tables 4.1 and 4.2 report the computational results for Datasets 1 and 2, respectively. The first three columns describe the instance characteristics: the time horizon T, the neighborhood size *radius*, the *ratio_range* (computed by $\#customers/\#stations$, where, for example, (0, 15] indicates ratios up to and including 15), and the number of instances with these specifications *count*. The next three columns report the performance of each heuristic, including the average running time in seconds ($time(s)$), the average percentage of customers served relative to the total number of customers in each instance ($served(\%)$), and the average optimality gap in percentage ($gap(\%)$). The last column reports the benchmark performance obtained by the MILP Formulation (4.1).

Focusing first on the effect of time horizon values in Table 4.1, we observe distinct trends across the three heuristics. When the time horizon is short ($T = 60, 240$), greedyHCapacity achieves the highest percentage of customers served and maintains zero optimality gaps, outperforming the other two approaches. However, as the time horizon increases ($T = 480, 720$), its running time grows significantly, and its performance declines. In contrast, greedySimpleCapacity performs best under longer time horizons, achieving the highest percentage of customers served and the smallest gaps. The Hybrid MILP heuristic improves its performance in optimality gaps as T increases from 60 to 480, after which its performance stabilizes.

These trends are expected. A longer time horizon allows more stations to be visited, leading to a larger h value in greedyHCapacity, which increases the time required to select candidate solutions H_1 and H_2. The 1.5-hour time limit restricts greedyHCapacity from fully exploring the solution space, limiting its ability to find the best solution. By comparison, greedySimpleCapacity follows a straightforward solution procedure, iteratively choosing the next station with the highest ratio that meets the deadline constraint. This simplicity keeps its computational time low. For the Hybrid MILP heuristic, which first solves a no-deadline max flow problem followed by a total distance minimization, instances with longer time horizons increase the likelihood that the travel time remains

within the deadline T, reducing the need to execute the second formulation multiple times. As a result, Hybrid MILP serves more customers and achieves shorter running times as T increases. However, beyond a certain point, such as when T increases from 480 to 720, the optimality gap no longer improves. This may be due to the expanded solution space associated with longer time horizons, which allows more feasible station combinations but does not guarantee the selection of the solution that serves the most parcels. Since instances with the same number of customers and stations are evaluated across different time horizons, we observe a notable impact on Hybrid MILP's performance. Meanwhile, greedySimpleCapacity shows consistent performance across different time horizons, with small variation. When looking at the ratio_range, all three heuristics and the MILP Formulation (4.1) serve fewer parcels as the ratio increases due to limited station capacities. Regarding the radius, a larger radius leads to a higher percentage of parcels served, as parcels have more station options.

To summarize, greedyHCapacity achieves best objective values when the time horizon is short, greedySimpleCapacity is fastest and maintains a stable performance across different settings, and the Hybrid MILP heuristic shows a competitive performance for longer time horizons. Additionally, while the worst-case performance of the uncapacitated version of greedyHCapacity is better than that of greedySimpleCapacity, under certain conditions where the time horizon is long or the running time is limited, greedySimpleCapacity can be a more practical choice.

We now analyze Table 4.2. In contrast to the previous tests, which consider a varying number of stations (ranging from 5 to 25), these tests always use the same set of 12 stations. Across all instances, the optimality gaps for the heuristics remain within 7% on average, with greedyHCapacity achieving zero optimality gaps. Similar to previous findings, as the time horizon increases, the running time of greedyHCapacity also increases, but it remains within an average of 10 minutes per instance. In contrast, greedySimpleCapacity maintains a consistently low running time, while Hybrid MILP becomes more efficient, reducing its computation time as the time horizon increases. Since each instance consists of 12 stations, the problem size remains manageable for greedyHCapacity, allowing it to reach optimal solutions. For Hybrid MILP, the average optimality gaps decrease slightly as T increases from 60 to 180. Looking at the ratio_range, we observe a trend consistent with the previous table: as the ratio increases,

all heuristics as well as the MILP model serve a smaller percentage of parcels due to capacity limitations. In addition, the running time increases with higher ratios, as more parcels must be evaluated during the selection process, particularly for greedyHCapacity, which requires more time to update candidate solution H_2.

On average, greedyHCapacity serves most customers when the number of stations that can be visited is limited. Although its computational time increases, it remains manageable with 12 stations. greedySimpleCapacity maintains consistent performance across different instance scales, making it a reliable choice, especially for larger problem sizes. In contrast, Hybrid MILP performs well on instances with longer time horizons while also offering relatively fast computation times.

Table 4.1: Computational Results with Dataset 1

				greedySimpleCapacity			**greedyHCapacity**			**Hybrid MILP**			**MILP (1)**	
T	radius	ratio_range	count	time(s)	served(%)	gap(%)	time(s)	served(%)	gap(%)	time(s)	served(%)	gap(%)	time(s)	served(%)
60	0	(0,15]	48	0.01	13.63	1.26	1.49	13.77	0.00	1.34	13.43	5.20	0.11	13.77
		(15,25]	42	0.01	10.17	0.49	2.92	10.21	0.00	2.62	9.7	8.25	0.21	10.21
		AVG/SUM		0.01	12.02	0.90	2.16	12.11	0.00	1.94	11.69	6.62	0.16	12.11
	10	(0,15]	48	0.01	14.38	0.61	2.07	14.43	0.00	1.48	13.85	7.78	0.11	14.43
		(15,25]	42	0.01	10.38	0.75	4.09	10.46	0.00	2.85	9.66	10.64	0.21	10.46
		AVG/SUM		0.01	12.52	0.67	3.01	12.58	0.00	2.12	11.89	9.11	0.16	12.58
	20	(0,15]	48	0.01	17.93	1.30	3.62	18.17	0.00	2.21	16.6	10.98	0.11	18.17
		(15,25]	42	0.01	11.23	0.60	7.28	11.31	0.00	3.69	9.43	21.15	0.21	11.31
		AVG/SUM		0.01	14.80	0.98	5.33	14.97	0.00	2.90	13.25	15.73	0.16	14.97
			AVG	**0.01**	**13.11**	**0.85**	**3.50**	**13.22**	**0.00**	**2.32**	**12.28**	**10.49**	**0.16**	**13.22**
240	0	(0,15]	48	0.01	39.59	1.65	121.80	40.09	0.00	1.03	39.25	2.98	0.10	40.09
		(15,25]	42	0.02	28.16	1.90	222.74	28.68	0.00	2.04	27.63	4.03	0.22	28.68
		AVG/SUM		0.01	34.26	1.77	168.91	34.77	0.00	1.50	33.83	3.47	0.16	34.77
	10	(0,15]	48	0.01	39.62	1.69	149.52	40.21	0.00	1.11	39.34	2.92	0.11	40.21
		(15,25]	42	0.02	27.18	2.72	280.88	28.03	0.00	2.23	27.09	4.32	0.22	28.03
		AVG/SUM		0.01	33.82	2.17	210.82	34.53	0.00	1.63	33.62	3.57	0.16	34.53
	20	(0,15]	48	0.01	49.38	1.48	214.94	50.03	0.00	1.72	48.58	3.32	0.15	50.03
		(15,25]	42	0.02	39.54	3.17	434.21	40.67	0.00	2.85	38.97	5.01	0.25	40.67
		AVG/SUM		0.02	44.79	2.27	317.27	45.66	0.00	2.25	44.10	4.11	0.19	45.66
			AVG	**0.01**	**37.62**	**2.07**	**232.33**	**38.32**	**0.00**	**1.79**	**37.18**	**3.72**	**0.17**	**38.32**
480	0	(0,15]	48	0.01	56.70	0.65	1880.42	55.54	3.98	0.78	55.92	2.46	0.10	57.01
		(15,25]	42	0.01	45.68	1.19	2287.63	43.91	7.05	1.5	45.08	2.74	0.21	46.19
		AVG/SUM		0.01	51.56	0.90	2070.45	50.11	5.41	1.12	50.86	2.59	0.15	51.96
	10	(0,15]	48	0.01	60.00	0.99	1921.68	58.86	4.18	0.84	59.57	2.05	0.11	60.55
		(15,25]	42	0.02	43.62	1.47	2321.15	41.49	8.04	1.65	43.14	2.94	0.22	44.35
		AVG/SUM		0.02	52.36	1.21	2108.10	50.75	5.98	1.22	51.90	2.47	0.16	52.99
	20	(0,15]	48	0.02	69.84	1.96	2006.73	69.74	2.19	1.19	69.09	2.98	0.18	71.11
		(15,25]	42	0.03	58.01	1.87	2406.95	55.42	7.58	2.09	57.58	2.92	0.28	58.99
		AVG/SUM		0.02	64.32	1.92	2193.50	63.06	4.71	1.61	63.72	2.95	0.22	65.46
			AVG	**0.02**	**56.08**	**1.34**	**2124.02**	**54.64**	**5.37**	**1.31**	**55.49**	**2.67**	**0.18**	**56.80**
720	0	(0,15]	48	0.01	65.26	0.46	2806.09	58.95	13.43	0.57	63.91	2.84	0.11	65.55
		(15,25]	42	0.02	52.96	0.52	2933.75	46.19	16.90	1.13	51.85	2.85	0.21	53.18
		AVG/SUM		0.01	59.52	0.49	2865.66	53.00	15.05	0.83	58.28	2.84	0.16	59.78
	10	(0,15]	48	0.01	69.69	0.50	2807.17	62.60	13.76	0.6	68.35	2.65	0.11	70.00
		(15,25]	42	0.03	58.28	0.70	2943.45	51.16	16.73	1.27	57.47	2.41	0.21	58.68
		AVG/SUM		0.02	64.37	0.60	2870.77	57.26	15.15	0.91	63.27	2.54	0.16	64.72
	20	(0,15]	48	0.02	83.14	1.21	2814.56	76.07	10.24	0.61	81.94	2.65	0.17	84.11
		(15,25]	42	0.04	69.11	1.77	2973.81	59.25	17.99	1.55	68.41	2.83	0.31	70.31
		AVG/SUM		0.03	76.59	1.48	2888.88	68.22	13.86	1.05	75.63	2.73	0.24	77.67
			AVG	**0.02**	**66.82**	**0.85**	**2875.10**	**59.49**	**14.68**	**0.93**	**65.73**	**2.71**	**0.18**	**67.39**
			AVG	**0.02**	**43.41**	**1.28**	**1308.74**	**41.42**	**5.01**	**1.59**	**42.67**	**4.90**	**0.17**	**43.93**

Table 4.2: Computational Results with Dataset 2

				greedySimpleCapacity			greedyHCapacity			Hybrid MILP			MILP (1)	
T	radius	ratio_range	count	time(s)	served(%)	gap(%)	time(s)	served(%)	gap(%)	time(s)	served(%)	gap(%)	time(s)	served(%)
60	0	(0,20]	153	0.00	37.57	3.47	6.97	38.90	0.00	0.54	38.01	2.36	0.06	38.90
		(20,40]	198	0.00	34.71	5.66	13.88	36.77	0.00	1.11	35.63	3.16	0.13	36.77
		(40,60]	149	0.01	32.38	3.74	21.31	33.66	0.00	1.72	32.77	2.64	0.19	33.66
		AVG/SUM	500	0.00	34.89	4.42	13.98	36.50	0.00	1.12	35.51	2.76	0.13	36.50
	3	(0,20]	153	0.00	58.24	1.95	7.28	59.44	0.00	0.53	58.48	1.64	0.08	59.44
		(20,40]	198	0.01	45.17	0.30	14.58	45.28	0.00	1.76	44.01	2.99	0.15	45.28
		(40,60]	149	0.01	33.40	2.50	22.27	34.27	0.00	2.92	32.13	6.28	0.21	34.27
		AVG/SUM	500	0.01	45.66	1.46	14.64	46.33	0.00	1.73	44.90	3.56	0.15	46.33
	5	(0,20]	153	0.00	72.98	1.30	7.73	73.98	0.00	0.53	72.29	2.37	0.10	73.98
		(20,40]	198	0.01	50.66	1.45	15.42	51.32	0.00	2.11	49.52	3.64	0.17	51.32
		(40,60]	149	0.02	34.91	1.85	23.32	35.60	0.00	3.16	33.93	4.90	0.22	35.60
		AVG/SUM	500	0.01	52.80	1.52	15.42	53.57	0.00	1.94	51.84	3.63	0.16	53.57
			AVG	**0.01**	**44.45**	**2.47**	**14.68**	**45.47**	**0.00**	**1.59**	**44.08**	**3.32**	**0.15**	**45.47**
120	0	(0,20]	153	0.00	57.28	2.89	57.17	58.98	0.00	0.39	57.44	2.62	0.06	58.98
		(20,40]	198	0.01	53.74	3.64	114.10	55.77	0.00	0.83	54.16	2.89	0.13	55.77
		(40,60]	149	0.01	49.48	2.73	175.63	50.88	0.00	1.33	49.63	2.44	0.19	50.88
		AVG/SUM	500	0.01	53.56	3.14	115.02	55.29	0.00	0.85	53.81	2.67	0.13	55.29
	3	(0,20]	153	0.00	90.58	0.15	61.75	90.70	0.00	0.16	89.78	1.01	0.08	90.70
		(20,40]	198	0.01	70.93	2.37	123.81	72.54	0.00	0.98	70.67	2.64	0.17	72.54
		(40,60]	149	0.02	51.05	2.79	185.18	52.58	0.00	2.15	50.76	3.37	0.21	52.58
		AVG/SUM	500	0.01	71.02	1.81	123.11	72.15	0.00	1.08	70.59	2.36	0.16	72.15
	5	(0,20]	153	0.00	92.10	1.80	66.18	93.78	0.00	0.15	91.54	2.36	0.11	93.78
		(20,40]	198	0.01	75.95	2.51	131.93	77.87	0.00	1.12	75.82	2.73	0.22	77.87
		(40,60]	149	0.03	52.64	1.37	194.39	53.42	0.00	2.35	52.02	2.58	0.23	53.42
		AVG/SUM	500	0.01	73.95	1.96	130.42	75.45	0.00	1.19	73.54	2.57	0.19	75.45
			AVG	**0.01**	**66.17**	**2.30**	**122.85**	**67.63**	**0.00**	**1.04**	**65.98**	**2.53**	**0.16**	**67.63**
180	0	(0,20]	153	0.00	76.79	2.31	146.92	78.59	0.00	0.21	77.51	1.39	0.07	78.59
		(20,40]	198	0.01	75.19	2.48	292.97	77.08	0.00	0.45	76.60	0.63	0.14	77.08
		(40,60]	149	0.01	72.20	1.27	447.30	73.14	0.00	0.71	72.98	0.22	0.21	73.14
		AVG/SUM	500	0.01	74.79	2.07	294.27	76.36	0.00	0.46	75.80	0.74	0.14	76.36
	3	(0,20]	153	0.00	99.27	0.03	157.82	99.29	0.00	0.09	95.45	3.88	0.09	99.29
		(20,40]	198	0.01	91.48	0.09	318.62	91.57	0.00	0.34	89.38	2.33	0.20	91.57
		(40,60]	149	0.02	76.58	0.00	474.42	76.58	0.00	0.98	75.20	1.80	0.23	76.58
		AVG/SUM	500	0.01	89.42	0.05	315.84	89.47	0.00	0.45	87.01	2.65	0.18	89.47
	5	(0,20]	153	0.00	100.00	0.00	167.43	100.00	0.00	0.12	93.43	6.57	0.07	100.00
		(20,40]	198	0.01	94.26	1.38	336.92	95.54	0.00	0.34	90.80	4.84	0.49	95.54
		(40,60]	149	0.03	78.35	0.61	498.11	78.89	0.00	0.98	77.45	1.82	0.24	78.89
		AVG/SUM	500	0.02	91.28	0.73	333.09	91.94	0.00	0.46	87.63	4.47	0.28	91.94
			AVG	**0.01**	**85.16**	**0.95**	**314.40**	**85.92**	**0.00**	**0.46**	**83.48**	**2.62**	**0.20**	**85.92**
			AVG	**0.01**	**65.26**	**1.91**	**150.64**	**66.34**	**0.00**	**1.03**	**64.51**	**2.82**	**0.17**	**66.34**

4.6 Conclusion

In this chapter, we propose a new problem in logistics, involving pickup stations and autonomous ground vehicles that can handle the shortage of workers. We show that the problem is NP-hard and analyze several variants that

reflect operational constraints commonly faced by shipping companies. To address this practically relevant yet unexplored problem, we develop algorithms inspired by classical combinatorial optimization problems, such as the 0/1 knapsack and the budgeted maximum coverage problems. Additionally, we derive performance guarantees for the proposed heuristics across problem variants and evaluate their average-case performance in the most challenging setting. Our extensive computational experiments demonstrate that the smallest optimality gaps obtained among the proposed heuristics remain within 3%, and greedyHCapacity achieves zero optimality gaps for the station number within 12 with an average running time under 10 minutes. Furthermore, our results highlight that the choice of solution approach should be tailored to the characteristics of each problem instance, as no heuristic dominates across all scenarios.

This study opens several directions for future research. A first and natural extension is the study with release dates in the same-day delivery setting, which was part of the original research goal. Analyzing the performance guarantees of these heuristics in such dynamic environments remains an open and promising research direction. Another possibility is to incorporate order-specific deadlines, where the heuristics developed in this chapter can be adapted to account for more consolidation opportunities.

5

Managing startup Growth Before Facility Investments

joint work with **Felix Papier** and **Riccardo Mogre**

5.1 Introduction

Scaling up is a critical phase in the growth journey of startups (Isenberg, 2012; Jules et al., 2022). It involves expanding their market presence while concurrently growing organizational structures and operational capabilities to source, produce, and deliver (Fine et al., 2022). However, scaling up becomes for many startups a chicken-or-egg problem: Achieving scale often requires substantial investments, for example in production capacity, while securing the necessary funding for such investments typically requires demonstrating the business potential of the startup, signaled by an already acquired customer base with recurring sales contracts (Soufani, 2001).

For example, Umiami, a food-tech startup known for its plant-based fillets, initially struggled to increase sales and secure fundings for production investment. After years of effort, the startup received $30 million Series A funding in April of 2022 (Vegconomist, 2022), enabling it to acquire a former Unilever factory in France to serve as its first commercial-scale production facility (The Plant Base, 2022). However, some entrepreneurs may attract high demands but be unprepared to scale the production to meet demands. As seen in startup cases such as Skinny-Girl Cocktails and Banza—introduced in Chapter 1.3—rapid growth without sufficient production preparation can lead to operational setbacks and missed opportunities for long-term value creation.

In the early stages, startups may accept negative product margins (as a consequence of inadequate production capacity and scale) in order to be able to build the critical mass of recurring sales needed for obtaining funding and being able to scale; prioritizing growth over profitability is common for startups in early growth phases. Beyond Meat, for example, prioritized growth over profitability in its early stages, focusing on market expansion despite operating with negative margins (Beyond Meat, Inc., 2024; Digital Food Lab, 2024).

Acquiring sales contracts too quickly, however, also comes with its inherent risks. In the example of SkinnyGirl and Banza, over-promising without the capability to deliver on their promises contributed to the startups' loss of control over their own operations. These cases highlight the difficulties startups are facing when scaling up business

too quickly. A sudden influx of new orders can result in substantial, more than proportional costs for adjusting to demand, such as costs for hiring and training workers (Pindyck, 1982), costs for reprocessing or compensating for non-conforming products and quality issues (Zhu et al., 2007), and treating rush orders. These costs are similar to what is known as adjustment costs for labor in economic theory, where these costs are identified as substantial when firms gradually change production levels in response to demand shocks (Hamermesh & Pfann, 1996). Such adjustment costs include the costs of disruption in production, retraining expenses, and inefficiencies that arise during the reallocation of resources and may discourage rapid capacity expansion. It is important to account for capacity adjustment costs when considering growth paths of startups.

One way to become quickly operational and to enter a market without significant investment into production capacity is to outsource production to contract manufacturers (CM). CMs may help quickly meet the expectations of early customers and investors (Steinbruch et al., 2022). As startups grow and customer demand becomes more predictable, entrepreneurs then acquire their own production facilities, a milestone that requires substantial investment. To attract the necessary investment for this transition, entrepreneurs have to demonstrate consistently growing demand or a strong customer base with recurring sales (Joglekar & Lévesque, 2009; Cavallo et al., 2019), which can be served with the CM's production capacity. In this paper, we use CM-based outsourcing of startups in the food-tech sector as motivational example, even though our anlysis also applies to other settings with negative margins.

In this study, we explore how startups can grow optimally, by balancing the costs of capacity adjustments with negative product margins when working towards a target size of recurring sales. Unlike established firms that can rely on existing resources to support new product launches, startups have to prove their business potential to be able to reach the next phase of development by acquiring a certain level of recurring sales, often even before being able to fulfill these sales orders sustainably. Growing too fast, by attracting a large number of new orders within a short time period, can lead to operational challenges such as compromised quality, overprotionally high costs for accelerated hiring and training of workers, treatment of rush order, and reprocessing defective products (see discussion

above), either internally or at contract manufacturers when production is outsourced. Conversely, growing too slowly risks missing market opportunities, as competitors establish stronger market positions and develop brand loyalty, and drain financial resources, as production is often no profitable at this stage of development of startups, making it more difficult for the startup to secure funding. Thus, the core challenge for startups is to find the right pace of growth: fast enough to meet market expectations and limit financial loss yet measured enough to avoid excessive costs for capacity adjustment costs related to too-fast growth.

While capacity expansion has been extensively studied for mature firms (Van Mieghem, 2003; Chao et al., 2009; Wang et al., 2013), limited research in operations management has focused on the unique challenges faced by startups with respect to capacity growth. As is common for start-ups, we take into account the trade-off between overproportional costs for fast-growing capacity with negative margins during the scale-up phase, and we explicitly consider the need, to obtain further funding, for developing a solid level of recurring sales through sales effort with uncertain outcome.

Specifically, we address the following research questions:

- What is the optimal growth path for a startup to reach its target level for a customer base with recurring sales? Should the growth be steady or slow down or accelerate at certain stages?
- How does uncertainty with respect to sales acquisition affect growth decisions? Does increased uncertainty encourage startups to grow faster or more cautiously?
- How do factors such as the capacity adjustment cost, the unit margin during the scale-up phase, and the target for recurring sales influence the optimal growth strategy?

To answer these questions, we develop a dynamic cost-minimization model that stops when either a sufficient number of recurring sales is acquired (the *target threshold*) or when the end of the horizon is reached. The amount

of recurring sales is the system state, which grows monotonically over time. The startup does not have cash constraints and decides in every period on the sales effort that it invests to acquire an (uncertain) number of new, recurring sales orders. We develop a dynamic cost-minimization model for a sequentially growing startup. We first analyze a deterministic setting, deriving closed-form solutions for the optimal sales effort in each period and the optimal number of periods required to reach the target. We then extend our analysis to a stochastic setting, where the number of new recurring sales acquired through sales efforts is uncertain.

Our results indicate that an optimal policy of intermediate growth exists that balances capacity adjustment costs with loss of margin. Specifically, sales effort should grow aggressively in early periods but slow down as the startup approaches the target threshold in later periods. Furthermore, total costs increase more than proportionally as the target increases, suggesting that startups should carefully assess their capabilities before committing to ambitious goals. When unit capacity adjustment costs are high, startups should grow more slowly, while the opposite is true for higher, negative unit margins. Finally, when sales acquisition is highly uncertain, startups benefit from a more cautious growth approach, postponing sales acquisition to later periods compared to a more predictable sales acquisition.

The rest of the chapter is organized as follows. Section 5.2 reviews the related works. In Section 5.3, we present the problem and its deterministic formulation. Section 5.4, analyzes the deterministic model and derives closed-form solutions along with managerial insights. In Section 5.5, we extend the model to a stochastic setting. In Section 5.6, we present numerical experiments and additional insights. We conclude with a summary and future research directions in Section 5.7. All proofs and technical details can be found in Appendix A.1.

5.2 Literature review

Our work builds on and contributes to three literature streams: entrepreneurial operations, dynamic capacity management, and outsourcing of manufacturing operations to CM. In the following, we briefly review the relevant research works within these domains.

Research on entrepreneurial operations: The objectives studied in the entrepreneurial literature can generally be categorized as milestone-based or continuous. Milestone-based objectives yield rewards or penalties only when a predefined milestone, target, or threshold is reached (as in this paper), whereas continuous objectives mean rewards or penalties accumulate over time. In the milestone-based stream, Babich & Sobel (2004) model a growing firm's operational and financial decisions up to an Initial Public Offering (IPO), maximizing the expected discounted IPO payoff. Considering the financial risk and potential bankruptcy while operating as a private firm, the main tradeoff is waiting to grow and risking bankruptcy versus going public early and securing a lower but safer payoff. In contrast, most of the literature considers continuous objectives. Specifically, Yoo et al. (2016a) model optimal time allocation policy for entrepreneurs, revealing that entrepreneurs should invest more in process improvement early on. Yoo et al. (2016b) study when entrepreneurs should hire their first employee and how their hiring differs from established firms, suggesting that early investment to save time in the future is worthwhile as entrepreneurs' time becomes more valuable with firm growth. Swinney et al. (2011) analyze differences in the timing of capacity investment between startups and established firms, considering that startups aim to maximize the probability of survival, while established firms seek to maximize expected profit. Lévesque et al. (2017) also explore the interplay between a startup and an established firm but they consider both a profit-maximizing startup and a survival-maximizing startup. Compared to Swinney et al. (2011), instead of focusing on capacity investment, Swinney et al. (2011) study the timing of a startup's production launch compared to that of an established company as entrepreneurs' time is usually severely constrained. Bhaskaran et al. (2021) study the tradeoff for cash-constrained startups between launching on-hand products to generate funds and the risk of cannibalizing

market for an advanced product. They find that startups might delay the introduction of on-hand products more when the risk of sudden failure of the advanced product is high. Another paper by Kagan et al. (2022) studies the strategies that contribute to successful entrepreneurs' fundraising. They find negotiating with multiple investors can have both negative and positive effects depending on elements such as investor size, substitutability, and the entrepreneur's outside options. The above papers show how entrepreneurs evaluate available options when facing various challenges. More recently, Wang et al. (2024) consider how an external advisor should recommend options to entrepreneurs. They find that the advisor may find it optimal to suggest inferior solutions before superior ones to motivate the entrepreneur to explore over time. While the above papers focus on improving entrepreneurs' performance subjecting to resource (time, financial, and human capital) constraints, Kundu et al. (2024) analyze the operating environments of small firms with a focus on reducing their downside losses due to disruptions. They note that both managerial and resource redundancies help firms build resilience against disruptions. Additional contributions also come from the broader management literature, for example, Tidhar et al. (2025) investigate how some new firms grow into mature firms and others do not. For a broader discussion of entrepreneurial research within and beyond the field of OM, we refer readers to the surveys (Lévesque & Stephan, 2020; Fine et al., 2022).

Most entrepreneurial OM research has focused on products, production process, inventory, or financing decisions. However, for startups to scale, a critical yet underexplored challenge is how to drive market growth to a target size while building production and delivery capacity. This chapter addresses that gap by developing a dynamic growth model in which a startup uses sales effort to build a customer base, secure funding for facility investment, and optimize its growth trajectory.

Research on dynamic capacity management: In this literature, capacity expansion is seen as an investment opportunity, often under stochastic demand. Van Mieghem (2003) provides a comprehensive review of strategic capacity management, including investment under uncertainty across three settings. Chao et al. (2009) characterize optimal expansion strategies involving immediate equipment purchases or deferred contracts, while Garcia & Shen (2010) develop a dynamic oligopoly model for energy infrastructure with stochastically growing demand.

Both papers above assume a single technology and risk-neutral decision maker. Extensions include competing technology investment options (Wang et al., 2013), risk-averse decision making (Lu & Yan, 2016), demand learning (Qi et al., 2017), and commodity price uncertainty (Bensoussan & Chevalier-Roignant, 2019). Another branch emphasizes the role of financial planning in successful expansion/growth. Churchill & Mullins (2001) propose a growth framework centered on internally generated revenue to reduce reliance on external funding, while Peura & Yang (2023) analyze how service platforms, like ride-hailing, can attracting new service providers who lack the necessary assets. Pisano (2024) emphasize that unbalanced growth–expanding faster than a firm's capabilities–can cause lasting damage to reputation and culture, as rapid growth leads to "growing pains" such as quality and customer service problems.

While literature has mainly focused on mature firms with ample resources and low risk of failure, little is known about how resource-constrained startups manage capacity growth to gain the trust of customers and investors. Our work addresses this gap by modeling startup sales effort as the driver of recurring sales growth, capturing growth pains (Pisano, 2024) through capacity adjustment costs that increase superlinearly with new sales.

Research on outsourcing of manufacturing operations to contract manufacturers: Another closely related literature stream explores the implications of outsourcing production to a supplier, specifically, a CM, within a supply chain context. Zhu et al. (2007) examine a scenario in which a buyer outsources production to a supplier, who may produce nonconforming units and incur capacity adjustment costs. Given that both the buyer and the supplier have incentives to improve quality, Zhu et al. (2007) investigate the roles of different parties in quality management. They find that only one party needs to make efforts to improve quality unless resource limitations exist. Gray et al. (2009) explore the effects of learning-by-doing on outsourcing operations given a powerful CM. Their study utilizes a two-period, game-theoretic model wherein both the original equipment manufacturer (OEM) and the CM can lower their production costs through learning-by-doing. They find that learning-by-doing can lead to dynamic changes in outsourcing strategy as the OEM and CM production costs evolve. Wang et al. (2021) consider two types of sourcing for a buyer: direct sourcing (DS)—the buyer acquires the component from the sup-

plier directly, and agent sourcing (AS)—the buyer authorizes the purchasing agent to carry on both acquisition and complementary services. They analyze how the bargaining power of three parties (buyer, DS, AS) affects the equilibrium procurement outsourcing structure and find a reservation profit spillover effect. Moreover, outsourcing the whole manufacturing of a product allows OEMs to free up capital, reduce labor costs, and concentrate on key activities such as R&D, design, and marketing. However, it also introduces the possible risk of the CM establishing its own brand and becoming a competitor to the OEM (Arruñada & Vázquez, 2006). Considering such a probability, a recent paper by Hou & Shen (2022) investigates the conditions that the CM will enter the end market and the consequence of such market entry for the OEM and the entire supply chain. Interestingly, they find that market entry will be less likely when the CM is relatively strong. Besides, there are also several recent papers exploring multiple outsourcing options, such as the choice between a competing manufacturer and a non-competing manufacturer (Luo et al., 2024), or horizontal outsourcing between competing duopolists (Hu et al., 2022). For more papers on outsourcing in supply chains, we refer to the comprehensive survey by Tsay et al. (2018).

Most work of this stream of literature focuses either on the outsourcing decision itself or on the management of the relationship between buyer and supplier (CM). We contribute to this stream, primarily investigating how to collaborate more effectively to reduce production costs, ensure quality, or maximize profit. In contrast, our paper shifts the focus to the startup itself. Assuming a stable outsourcing relationship at this stage, we study how startups can grow efficiently while relying on a CM and eventually transition to in-house production.

5.3 Model

In this section, we first describe the modeling assumptions in their business context in Section 5.3.1, before defining the model formally as a stochastic dynamic program in Section 5.3.2.

5.3.1 Problem Description

We consider a startup in the process of increasing its customer base through sales activities. Currently, the startup fulfills orders via a CM but is committed to scaling up by purchasing its own production plant. The startup faces a limited time frame to make this investment for two key reasons. First, from a market perspective, delays in scaling may enable competing firms to occupy the market, making it harder for the startup to expand in future. Second, from a funding perspective, startups often operate under strict timelines set by investors, particularly in milestone-based funding arrangements (Simeonova, 2024). In such arrangements, fundings are released in slices, with each slice released upon achieving specific milestones, such as hitting specific sales acquisition figures, which we consider in our analysis. In our model, these considerations are reflected in a finite planning horizon N, which represents the practical urgency to meet growth milestones under market or investors pressure. We investigate how the startup navigates its growth within this limited horizon to secure sufficient contracted sales to raise investment funds while maintaining quality and controlling costs.

The startup tracks the number of recurring sales received at time t, and based on this information, dynamically decides the sales effort to attract new recurring sales. Sales effort activities include meetings with potential customers, attendance at exhibitions and workshops, and advertising.

When the startup achieves a certain known level of recurring sales, it secures the necessary financing from investors to acquire a production site, supported by the notion that startups can attract investors with a solid customer base and proven sales track records.

At each period t before the production site is acquired, the startup employs a CM to fulfill sales up to t. The startup receives a profit that depends on the cumulative sales, which are recurring, given by the revenues associated with the sales minus the production costs incurred through the CM. Moreover, as previously introduced, the startup incurs capacity adjustment costs, which are determined by the period-to-period increase in its sales.

If the startup does not manage to reach the necessary recurring sales level for financing by the horizon N, it faces a positive terminal cost, representing real and reputational losses associated with missing the target. The end of the horizon N may correspond to an explicit deadline set by investors or a time beyond which securing financing would become significantly more difficult.

The startup aims to minimize the total costs over the horizon N, which include sales effort costs, production margins, and capacity adjustment costs, and the costs for falling short of the target.

5.3.2 Problem Formulation of the Deterministic Setting

We formulate the problem as a finite-horizon Markov Decision Process (MDP) with a continuous state space and continuous control space. The discrete time is indexed with $t = 1, 2, \ldots, N$.

The state of the system, denoted with x_t, is the number of recurring sales up to period t that the startup has to fulfill from that point onwards. We denote with k the level of recurring sales necessary to secure financing from investors. We assume that this level is deterministic and known by the startup in advance. Hence, the state space is the real interval $\mathcal{X} = [0, k]$, where $x_t = k$ indicates that the startup has purchased the production site.

At the beginning of period t, the startup observes x_t and decides the control u_t, which is the sales effort to attract new sales. The control space $\mathcal{U}(x_t)$ is defined as follows:

$$\mathcal{U}(x_t) = \begin{cases} [0, +\infty), & \text{if } 0 \leq x_t < k, \\ \{0\}, & \text{if } x_t = k. \end{cases} \tag{5.1}$$

The new sales received by the startup are denoted by $w_t(u_t)$. We assume a multiplicative element for w_t, i.e., $w_t(u_t) = z\,u_t$, where z is a positive scaling coefficient, such a linear model is commonly used in related literature

(see, e.g., Taylor, 2002).

The sales effort has a unit cost of $c > 0$, which incorporates both real and opportunity costs for the startup. Before the startup acquires the production site, sales orders are manufactured externally via CM. For each unit produced, the startup incurs a negative margin of $m < 0$, as the CM takes a share of the revenue, and startups often prioritize growth over immediate profitability.

While using CM, the startup also incurs capacity adjustment costs, which we model with the function $Q(\cdot)$. This function is assumed to be convex and increasing with respect to its argument, the number of new sales. This assumption reflects the fact that scaling production becomes increasingly costly as demand surges. For instance, a 2% increase in sales can often be managed relatively easily by utilizing existing raw materials and asking a few workers to work overtime. However, a 20% increase in sales typically requires more adjustments, such as hiring and training new workers, asking for quick delivery of raw materials, reprocessing unqualified products, which can disrupt production processes. These adjustments introduce costs that grow over proportionally, i.e., the cost of accommodating a 20% increase is significantly greater than ten times the cost of a 2% increase. After the startup buys its own plant, we consider the scaling-up process completed and assume no further costs or margins for subsequent periods.

We formulate the problem as a minimization problem where the startup aims to minimize, over the finite horizon, the difference between the sum of the costs (i.e., the sales effort costs, the capacity adjustment costs, and the applied terminal cost for missing the target) and the sum of the profits.

The transition equations for the problem are as follows:

$$x_{t+1} = \min(x_t + w_t(u_t), k) \qquad t = 1, 2, \ldots, N \tag{5.2}$$

The terminal cost $g_{N+1}(x_{N+1})$ represents the penalty incurred if the startup fails to reach the level of recurring sales k required to secure funding from investors. This cost reflects the consequences of not reaching the target, such as the need to pursue smaller or alternative investments, which can directly impact the startup's profitability. A notable example is Britishvolt, a UK-based startup focused on manufacturing lithium-ion batteries for electric vehicles (Wikipedia, 2025). Britishvolt began constructing a factory in Northumberland in 2021. However, the UK government refused to release £30M of its grant funding to Britishvolt because Britishvolt failed to meet specific milestones. Construction was halted in 2022, and the startup went to administration in 2023 followed by liquidation in 2024. Thus we use a penalty to capture these impacts in the terminal cost function. We assume that $g_{N+1}(x_{N+1})$ is a convex decreasing function, emphasizing that the penalty diminishes as the recurring sales approach the target[1].

The one-step cost function, or expected cost in a period, for period $t \leq N$ given state x_t and control u_t, is defined as:

$$g(x_t, u_t) = c\, u_t - m(x_t + w_t(u_t)) + Q(w_t(u_t)), \tag{5.3}$$

if $x_t < k$, and $g(x_t, u_t) = 0$ if $x_t = k$.

We aim to find a policy π that minimizes the total cost over N for a fixed initial number of recurring sales x_1:

$$\sum_{t=1}^{N} g(x_t, \pi_t(x_t)) + g_{N+1}(x_{N+1}), \tag{5.4}$$

where the cost accumulates only until the system reaches the target $x_t = k$, after which no further cost is incurred.

[1] An example of $g_N(x_N)$, where $p > 0$ is a parameter, for $0 \leq x_N \leq k$, is as follows: $g_N(x_N) = p\, k - p\, x_N$.

5.4 Analysis of the deterministic formulation

In the following, we analyze the deterministic model introduced in Section 5.3.2.

For simplicity and without loss of generality, we assume the scaling factor z in $w_t(u_t) = z \cdot u_t$ to be equal to one. In each period t, the decision maker uses a CM to serve the existing base of $\sum_{i=1}^{t} u_i$ sales at unit margin $m < 0$. Further, the acquisition of u_t new sales in period t leads to capacity adjustment cost $Q(u_t)$ at the CM for adjusting to the new volume.

Given a sufficiently long but finite planning horizon, there exists an optimal number of periods n^* for the startup to reach k. This leads to two possible cases:

- case A, Horizon Constraint Non-Binding ($n^* \leq N$): The startup reaches k within the available time horizon N;
- case B, Horizon Constraint Binding ($n^* > N$): The startup is unable to reach k within N periods.

When $n^* \leq N$, there will be no terminal penalty cost occurred, which means the analysis will be independent of the terminal penalty cost, and it can be omitted. However, when $n^* > N$, the terminal penalty cost will matter.

In case B, startups may choose not to reach the sales target k under time pressure and instead pay for the penalties for falling short. Nonetheless, it is still interesting to analyze the conditions under which reaching k is optimal versus when it is not, and to compare the associated costs when k is reached with those in case A. Figure 5.1 shows that costs incurred in case B are much higher than those in case A, particularly when fewer periods are available in case B, which means reaching k quickly is more costly. Additionally, in case A, it shows that there is no benefit to prolonging the process beyond the optimal number of periods n^* to reach k.

In the following, we will analyze case A first.

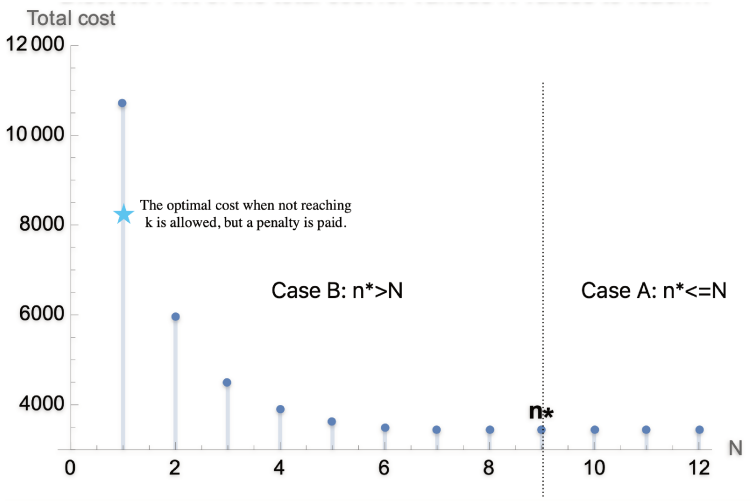


Figure 5.1: Illustration of total costs incurred in the two cases, assuming the startup reaches k within N periods, with parameters $k = 100, m = -5, q = 1$, and $c = 2$. The star indicates the optimal cost when $N = 1$ and failing to reach k is allowed with a penalty, given by $g_N(x_N) = p\,k - p\,x_N$, where $p = 107$.

5.4.1 Case A, Horizon Constraints Non-Binding ($n^* \leq N$):

Model with exactly n periods. For the sake of analysis, let us first assume that the startup manages to reach the necessary level of recurring sales for financing by the end of n, i.e., $x_n = k, n \leq N$. In the last period n, the decision maker applies remaining sales effort $u_n = k - \sum_{t=1}^{n-1} u_t$. We require $u_n \geq 0$, i.e., $\sum_{t=1}^{n-1} u_t \leq k$.

The above assumptions lead to the total costs of

$$\begin{aligned} C_n(\bar{u}) &= \sum_{t=1}^{n} Q(u_t) - m \cdot \sum_{t=1}^{n} \sum_{i=1}^{t} u_i + c \cdot k \\ &= \sum_{t=1}^{n-1} Q(u_t) + Q(k - \sum_{t=1}^{n-1} u_t) - m \cdot \sum_{t=1}^{n-1} (n-t)u_t - m \cdot k + c \cdot k, \end{aligned} \tag{5.5}$$

where $\bar{u} = (u_1, ..., u_{n-1})$. The resulting optimization problem can be summarized as

$$\min_{\bar{u}} C_n(\bar{u}), \text{subject to } u_t \geq 0, \text{ for } t = 1...n-1, \text{ and } \sum_{t=1}^{n-1} u_t \leq k. \tag{5.6}$$

Function $Q(\cdot)$ is convex by assumption, and the composition of a linear function and a convex function is also convex (Proposition 2.1.3 of Simchi-Levi et al. (2005)). Hence, function $C_n(\bar{u})$ is (jointly) convex in u_t. Since the constraints are also convex, the optimization problem (5.6) is convex (Section 4.2.1, page 136 of Boyd & Vandenberghe (2004)). Let us denote its solution by $\hat{u}^*$.

First, let us solve the problem in the absence of constraints, i.e., we minimize function $C_n(\bar{u})$. We refer to this problem as the *unconstrained* problem. From convexity follows that the optimal solution $\bar{u}^*$ is defined by the first-order conditions

$$\frac{\partial C_n}{\partial u_t} = 0 \Leftrightarrow Q'(u_t) - Q'(k - \sum_{i=1}^{n-1} u_i) - m(n-t) = 0, \forall t = 1...n-1. \tag{5.7}$$

If m were zero, the optimal solution would be $u_t = k - \sum_{i=1}^{n-1} u_i$ and, consequently, $u_t = k/n$. However, because $m < 0$, this solution is not optimal anymore.

In the next proposition, we provide the structure of the optimal solution to the problem (5.6), i.e., with constraints and with a fixed number of periods n (see the proof of Proposition 5 in Appendix A.1).

Proposition 5. *(a) $u_t^* < 0$ implies $\hat{u}_t^* = 0$, i.e., if the optimal sales effort in period t is negative in the unconstrained problem, the non-negativity constraint will be binding at the optimal solution of the constrained problem.*
(b) Let us assume that $\hat{u}_1^* = 0$. Then, the optimal solution to problem (5.6) is given by

$$\hat{u}^* = (0, \hat{u}^* | n-1), \tag{5.8}$$

where $\hat{u}^*|n-1$ *is the optimal solution to the problem with* $n-1$ *periods.*

(c) It holds $u_t^* \leq u_{t+1}^*$ *for all* $t = 1...n-1$*, i.e., the optimal unconstrained sales volumes are increasing in period t.*

(d) $u_1^*|n-1 \leq 0$ *implies* $u_1^*|n \leq 0$*, i.e., if in the unconstrained problem with* $n-1$ *periods, the optimal sales effort is non-positive in the first period, the same holds for all problems with n periods (or more).*

The results of Proposition 5 provide some insight into the structure of the optimal policy. Part (a) indicates that we can conclude from a negative optimal sales effort in the unconstrained solution that a zero sales effort is optimal for the final, constrained solution. Part (b) shows that if the first sales effort is zero in the optimal solution, the optimal sales efforts of the following periods are identical to those of a problem with a shorter horizon. Part (c) establishes that the optimal levels of sales effort are increasing over time. Parts (a)-(c) together imply part (d) that, as soon as the first optimal sales effort is zero, it will stay zero for any earlier periods that are added to the problem. Part (d) is a key property that we will employ in the next paragraph for determining the *optimal* number of time periods in which to reach target level k.

Optimization of the number of periods. Next, we determine the optimal number of periods, n^*. This allows us to derive the optimal solution to our original problem (with an unknown number of periods) from the pair n^* and $\hat{u}^*|n^*$.

We establish the following theorem by using the results of Proposition 5.

Theorem 3. *The optimal number of periods,* n^**, that minimizes* $C_n(\hat{u}^*|n)$*, is given by*

$$n^* = \min\{n|(u_1^*|(n+1) \leq 0)\}, \tag{5.9}$$

and the optimal sales efforts $\hat{u}_t^*$ *are defined by the* $(n-1)$ *conditions in* (5.7)*.*

Theorem 3 reveals that pursuing the fastest possible growth, such as reaching k in a single period, may not be

optimal due to the associated high capacity adjustment costs. Instead, startups should adopt a strategic growth path that balances financial loss from negative margins for serving already acquired sales with the cost to adjust the production capacity to the growth trajectory.

As illustrated in Figure 5.1, the costs incurred from a suboptimal growth approach, such as attempting to reach k in one step, are significantly higher compared to a more strategically planned growth path. As mentioned before, this happened to SkinnyGirl Cocktails when they grew too fast to meet the production and distribution capacity, and then they were sold with a discounted price in 2021. Thus it is important for startups to evaluate growth strategies with a focus on minimizing costs and enhancing long-term sustainability (see the proof of Theorem 3 in Appendix A.1).

Quadratic capacity adjustment cost. Next, we analyze a special case with a *quadratic* adjustment cost function, $Q(u) = q \cdot u^2$ where $q > 0$ is a constant. In this case, we can obtain a closed-form solution to the problem. With this quadratic capacity adjustment cost, condition (5.7) can be written as

$$\frac{\partial C_n}{\partial u_t} = 0 \Leftrightarrow 2qu_t^* - 2q(k - \sum_{i=1}^{n-1} u_i^*) - m(n-t) = 0, \forall t = 1...n-1. \tag{5.10}$$

Summing up the $n-1$ equations leads to

$$\begin{aligned}
&2q\sum_{t=1}^{n-1} u_t^* - 2q\sum_{t=1}^{n-1}(k - \sum_{i=1}^{n-1} u_i^*) - \sum_{t=1}^{n-1} m(n-t) = 0 \\
\Leftrightarrow\ &2q\sum_{t=1}^{n-1} u_t^* - 2q(n-1)(k - \sum_{t=1}^{n-1} u_t^*) - (n-1)mn + m\sum_{t=1}^{n-1} t = 0 \\
\Leftrightarrow\ &2q(1+n-1)\sum_{t=1}^{n-1} u_t^* - (n-1)mn + m\frac{n(n-1)}{2} - 2q(n-1)k = 0 \\
\Leftrightarrow\ &\sum_{t=1}^{n-1} u_t^* = \frac{(n-1)(mn+4qk)}{4qn}
\end{aligned} \tag{5.11}$$

for the sum of sales efforts and, by re-inserting this solution into each individual equation, we obtain

$$2qu_t^* - 2q(k - \sum_{i=1}^{n-1} u_i^*) - m(n-t) = 0$$
$$\Leftrightarrow u_t^* = \frac{mn + mn^2 - 2mnt}{4nq} + \frac{k}{n}, \forall t = 1...n-1. \tag{5.12}$$

The optimal number of periods, n^*, can be derived from

$$u_1^* = \frac{mn^* + mn^{*2} - 2mn^*}{4n^*q} + \frac{k}{n^*} < 0 \tag{5.13}$$

i.e.,

$$mn^* + mn^{*2} - 2mn^* + 4kq < 0, \tag{5.14}$$

which leads to

$$n^* = \min\left\{ n | (n+1) > \frac{1}{2} + \frac{1}{2}\sqrt{\frac{m - 16kq}{m}} \right\}, \tag{5.15}$$

such that the optimal solution to the problem with quadratic capacity adjustment cost is given by (5.12) and (5.15).

Insights from the quadratic capacity adjustment cost

From the closed-form solution for u_t in (5.12), we can infer that the optimal sales effort for each period is not monotone in k, as n^* also changes. In Figure 5.2, we see that when k increases from 35 to 80, u_t^* decreases while the time period to reach the higher investment level increases; when k increases from 80 to 125, u_t^* increases and the time period increases. This implies that, in a startup context, even a slight change in financing can significantly alter the growth strategy. For example, it may be better to pursue slower growth with a longer time period to reach a higher investment level.

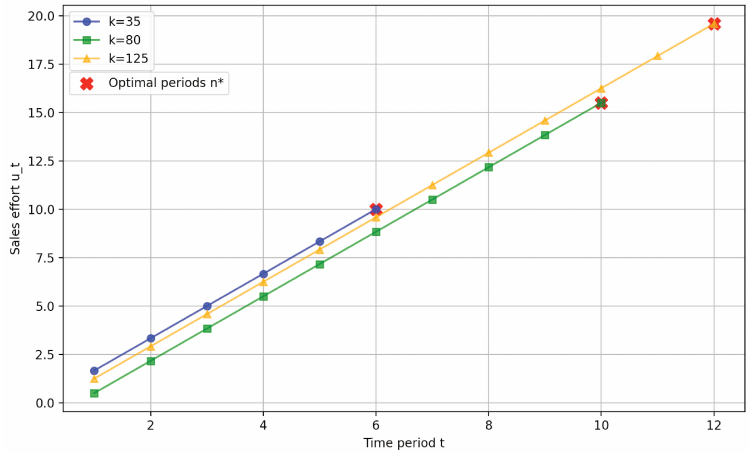


Figure 5.2: The changes of growth trajectories for different target k ($m = -10, q = 3, c = 3$).

From (5.12), we can observe the following derivative:

$$\begin{aligned} \frac{du_t^*}{dt} &= \frac{d}{dt}\frac{mn + mn^2 + 4kq - 2mnt}{4nq} \\ &= \frac{-m}{2q} > 0 \end{aligned} \tag{5.16}$$

This result shows that the optimal sales effort, u_t^*, increases linearly in time t, assuming constant parameters $m < 0$ and quadratic capacity adjustment cost given by $Q(u) = q \cdot u^2, q > 0$. The increase in sales effort over time reflects higher costs associated with early-stage growth, as the same customer base needs to be served throughout the entire period and the negative margins resulted must be covered. As a result, the entrepreneur should steadily increase the growth speed.

Proposition 6. *The optimal number of periods n^* is non-decreasing in k.*

The proposition reflects that the scale of funding goals critically influences the timing and pace of startup growth. When a startup sets smaller funding goals (k), it is better to grow more quickly over fewer periods to establish an in-house facility. Conversely, when pursuing larger funding goals, startups may find it optimal to

spread growth over more periods to preserve operational stability (reduce capacity expansion costs). For example, Daily Dose, a UK-based cold-pressed juice company, demonstrates this principle. Initially, the company pursued a small funding goal via crowdfunding, raising £125k (Crowdcube, 2018). This enabled them to rapidly expand their site from 500 sq ft to 2500 sq ft, allowing for quick market entry and early control over its supply chain. As their business matured, it pursued bigger funding goals to support further expansion plans. This phenomenon can also be observed in Figure 5.2, where n^* increases as k rises from 35 to 125 (see the proof of Proposition 6 in Appendix A.1).

Proposition 7. *The optimal number of periods n^* decreases in $|m|$ ($m < 0$).*

Proposition 7 suggests that as the CM production cost increases (a larger $|m|$ represents lower margins, which corresponds to higher CM costs), the company should reduce the reliance on CM and transition to in-house production more quickly. When the cost of outsourcing to CM is high, shifting to internal production becomes more economical, as prolonged dependence on external manufacturing leads to higher overall costs.

This result aligns with the notion that in-house production, despite its fixed or perhaps higher initial costs, could be more cost-efficient in the long run when compared to the escalating costs associated with CM, especially in scenarios where long-term reliance on external manufacturing partners is not sustainable (see the proof of Proposition 7 in Appendix A.1).

Next, let us examine how the total cost changes as the target recurring sales k varies, reflecting different investment levels, using numerical examples. In Figure 5.3, we observe that the total cost increases as the target rises, and the increase is more than proportional, which means that costs grow at an accelerating rate as k increases. This more than proportional increase in cost suggests that startups face increasing challenges when aiming for higher target levels. This phenomenon is reasonable as the target increases, more resources are required, including direct financial expenditure, effort, and time. Therefore, startups must carefully evaluate their capabilities and the potential return on investment. Before committing to ambitious targets, it is crucial to fully understand the cost

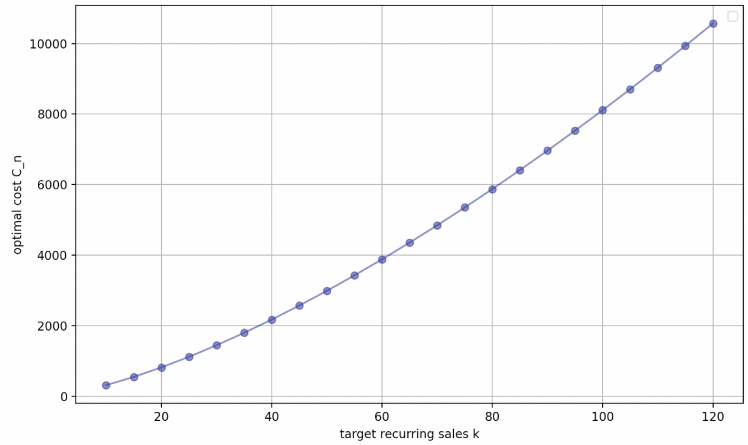


Figure 5.3: Optimal cost increasing in target recurring sales k (other parameters $m = -10$, $q = 3$, and $c = 3$).

implications.

Here, we assume that the unit cost of sales effort is static. However, in practice, it may decrease over time as the startup builds market reputation and gains customer trust, making it easier and less costly to attract new sales. Consequently, a reduction in sales effort cost could lead to an overproportional increase in sales effort over time, as the startup capitalizes on improved cost efficiency to accelerate growth.

5.4.2 Case *B*, Horizon Constraints Binding ($n^* > N$):

In this case, where startups operate under time pressure, they need to decide whether to reach the sales target k within a limited planning horizon or to accept penalties for falling short. If reaching k within a short time is too expensive due to high capacity adjustment costs, and the penalty for not reaching is relatively low, it may be more economical not to reach the target. In such cases, startups might opt for smaller investments or not buy a factory at that time.

To analyze this decision, we evaluate the last period cost assuming the initial state at the final period N is x_N.

The firm chooses the effort u_N to minimize the final period cost:

$$\min_{u_N} g_N = Q(u_N) + c \cdot u_N - m \cdot (x_N + u_N) + p(k - x_N - u_N), \text{ subject to } 0 \leq u_N \leq k - x_N. \tag{5.17}$$

For the unconstrained version, from convexity follows that the optimal effort u_N^* is defined by the first-order conditions

$$Q'(u_N^*) + c - m - p = 0. \tag{5.18}$$

Solving for u_N^*, we get

$$u_N^* = (Q')^{-1}(m - c + p). \tag{5.19}$$

With the linear constraints added, the problem is still convex, $\hat{u}_N^* = min([(Q')^{-1}(m - c + p)]^+, k - x_N)$. Assuming $(Q')^{-1}(m - c + p) > 0$ for having a well-posed optimization problem, we have

$$\hat{u}_N^* = \begin{cases} (Q')^{-1}(m - c + p), & \text{if } x_N \leq k - (Q')^{-1}(m - c + p) \\ k - x_N, & \text{otherwise.} \end{cases}$$

When the future effort needed to reach the target is relatively small, i.e., when $k - (Q')^{-1}(m - c + p) < 0$, the startup will optimally choose to impose $\hat{u}_N^* = k - x_N$ in the final period N to reach k . In other words, if the penalty p for missing the target exceeds a certain threshold, $p > \frac{k}{(Q')^{-1}} - m + c$, then reaching the target becomes the optimal decision, and it is optimal to put $k - x_N$ sales efforts to reach k.

Now assuming this condition holds and the startup chooses $\hat{u}_N^* = k - x_N$, the final period cost becomes:

$$\min_{u_N} g_N = Q(k - x_N) + c \cdot (k - x_N) - m \cdot k. \tag{5.20}$$

To analyze the total cost over the entire periods, we define the cumulative cost from periods 1 to $N-1$ as $f(\bar{u})$, $\bar{u} = (u_1, ..., u_{N-1})$

$$f(\bar{u}) = \sum_{t=1}^{N-1} Q(u_t) - m \cdot \sum_{t=1}^{N-1} (N-t)u_t + c \cdot \sum_{t=1}^{N-1} u_t. \tag{5.21}$$

Then the total cost over the entire periods is

$$\begin{aligned} C_N(\bar{u}) &= f(\bar{u}) + g(\bar{u}) \\ &= \sum_{t=1}^{N-1} Q(u_t) + Q(k - x_N) - m \cdot \sum_{t=1}^{N-1} (N-t)u_t - m \cdot k + c \cdot \sum_{t=1}^{N-1} u_t + c \cdot (k - x_N) \end{aligned} \tag{5.22}$$

As $x_N = \sum_{t=1}^{N-1} u_t$, it is exactly the equation form of (5.5) with $n = N$:

$$C_n(\bar{u}) = \sum_{t=1}^{n-1} Q(u_t) + Q(k - \sum_{t=1}^{n-1} u_t) - m \cdot \sum_{t=1}^{n-1} (n-t)u_t - m \cdot k + c \cdot \sum_{t=1}^{n-1} u_t + c \cdot (k - \sum_{t=1}^{n-1} u_t).$$

This confirms that to minimize $C_N(\bar{u})$, we can use the previous derived characterization of the optimal final-period decision $\hat{u}_N^*$.

Proposition 8. *Consider a quadratic capacity adjustment cost $Q(u) = qu^2$, with $q > 0$, and assume $\frac{m-c+p}{2q} > 0$. Then the optimal sales effort in the final periord N $\hat{u}_N^*$, is given by:*

$$\hat{u}_N^* = \begin{cases} \frac{m-c+p}{2q}, & \text{if } \frac{(N-1)(mN+4qk)}{4qN} \leq k - \frac{m-c+p}{2q} \\ k - \frac{(N-1)(mN+4qk)}{4qN}, & \text{otherwise.} \end{cases}$$

To illustrate this more concretely, consider the case of a quadratic capacity adjustment cost $Q(u) = qu^2$, with $q > 0$. Proposition 8 helps determine under what conditions the startup will commit sufficient sales effort in the

final period to reach the sales target. In particular, when the gap $k - \frac{m-c+p}{2q}$ is negative, the startup is guaranteed to reach the target k in period N by putting sales effort $\hat{u}_N^* = k - \frac{(N-1)(mN+4qk)}{4qN}$. If the penalty for not reaching is sufficiently high, i.e., if $p > 2kq - m + c$, then it is optimal for the startup to make a final push with $k - \frac{(N-1)(mN+4qk)}{4qN}$ units of sales efforts to reach the target k. This implies that when the cost of missing the target outweighs the burden of added effort, startups should prioritize reaching the target even with a tight time horizon (see the proof of Proposition 8 in Appendix A.1).

To summarize, when horizon constraints are binding, startups face a trade-off between reaching the target and accepting a penalty for falling short. If the penalty is sufficiently high (i.e., $p > 2kq - m + c$), it becomes optimal to reach the target on time to secure the facility investment. Otherwise, startups may opt for smaller investments or delay the acquisition of a factory.

5.5 Extension to a stochastic setting

The model discussed above assumes a stable market condition in which sales are predictable. However, in many real-world scenarios, market conditions can be volatile, making the outcomes of sales efforts uncertain. To account for this, we extend the deterministic model described in Section 5.3.2 to a stochastic setting, where the units of new sales generated by the startup's sales efforts are modeled as random variables $W_1(u_1)$, $W_2(u_2)$, ..., $W_N(u_N)$. Each random variable $W_t(u_t)$ is assumed to be positive, continuous, and to have finite mean and variance, increasing with the level of sales effort u_t. A realistic model that characterizes the relationship between W_t and u_t is the linear model $W_t(u_t) = u_t Z_t$, where Z_t is the realization at time t of a positive continuous random variable Z with mean μ and density function $\varphi(\cdot)$, assumed to be independent over periods.

The one-step cost function, or expected cost in a period, for period $t \leq N$ given state x_t and control u_t, is defined

as:

$$g(x_t, u_t) = c\,u_t - m\int_0^{\frac{k-x_t}{u_t}} (x_t + u_t\,z)\varphi(z)dz + \int_0^{\frac{k-x_t}{u_t}} Q(u_t\,z)\,\varphi(z)dz \tag{5.23}$$

if $x_t < k$, and $g(x_t, u_t) = 0$ if $x_t = k$.

We aim to find a policy $\pi_1, \pi_2, \ldots, \pi_N$ that minimizes the total expected cost over N for a fixed initial number of recurring sales x_1:

$$\underset{Z_1, Z_2, \ldots, Z_N}{\mathbb{E}} \left\{ \sum_{t=1}^{N} g(x_t, \pi_t(x_t)) + g_{N+1}(x_{N+1}) \right\}. \tag{5.24}$$

We denote with $J_t(x_t, u_t)$ the cost-to-go at time t, and with $J_t^*(x_t)$ the optimal cost-to-go at time t. For this problem, we have the following optimality equations, for $t = 1, 2, \ldots, N$:

$$J_t^*(x_t) = \min_{u_t \geq 0} c\,u_t + \int_0^{\frac{k-x_t}{u_t}} \left[-m(x_t + u_t\,z) + Q(u_t\,z) + J_{t+1}^*(x_t + u_t\,z) \right] \varphi(z)dz \tag{5.25}$$

and $J_{N+1}^*(x_{N+1}) := g_{N+1}(x_{N+1})$, for each x_N.

The formulation above is an extension of the deterministic model to account for uncertainty in sales outcomes. While we do not derive analytical results for this case due to its complexity, we implement the model in numerical experiments (see Section 5.6.3) to examine how different levels of sales uncertainty influence a startup's growth strategy. We also compare results under different levels of convexity in the capacity adjustment cost function, highlighting key differences between deterministic and stochastic settings.

5.6 Numerical experiments

In this section, we analyze how the growth trajectory is influenced by the convexity of the capacity adjustment cost function and the stochasticity of new sales acquisitions, using parameters calibrated from plant-meat companies.

We begin by describing parameter settings for a base case under a deterministic scenario and examining its resulting growth trajectory. Next, we investigate the impact of varying convexity levels in capacity adjustment costs. Finally, we extend the analysis to a stochastic setting, where new sales from sales efforts are uncertain.

5.6.1 Parameter Description and Preliminary Analysis

We assume that one period corresponds to one month. In October 2023, Umiami secured €32.5 million in investment capital to support the development of its production facility in the Alsace region. At that time, approximately over the preceding three years, the company had acquired €45 million in pre-sales (Businesscoot, 2022), equivanlent to an average of around $1.3 million in recurring monthly sales[2]. Our analysis focuses on understanding how Umiami reached this level of $1.3 million.

Due to the absence of publicly available cost data from Umiami, we take Beyond Meat, a company with similar product offering, as a benchmark (Beyond Meat, Inc., 2021). Prior to investing in its second production facility in Columbia, Missouri, in 2018, Beyond Meat reported revenues of $32 and $16 millions in years of 2017 and 2016, respectively. Over the same period, the company had gross losses of $2 millions in 2017 and $6 millions in 2016. Additionally, its selling and marketing expenses for attracting new customers and retaining existing ones, are at around $17 millions in 2017 and $12 millions in 2016. Based on these data, we estimate the unit margin m as -$0.17 per dollar of sales by taking the average net loss per sale in years 2016 and 2017. Similarly, the unit cost of sales effort c is set as $0.61 per dollar of sales, derived from the average selling and marketing expenses per dollar of sales over the same period.

Next, we determine parameters of the capacity adjustment cost function $Q(u) = q \cdot u^a$, which captures the additional costs incurred when adapting the production process to meet increased demand. These costs include

[2] Exchange rate assumed: €1 = $1.04 at the time of writing

expenses such as for hiring and training new workers and are assumed to grow overpropotionally with the size of new demands. This general power function covers several cases: it includes the quadratic special case when $a = 2$, and allows for more convex forms when $a > 2$, and less convex ones when $a < 2$. Since exact production cost data for each segment are unavailable, we infer the parameters based on the following assumptions. First, we assume that the capacity adjustment cost required to reach a production level of k within one year accounts for 5% of total production costs. Given an average monthly sales volume of \$1.3 million and a unit margin of -\$0.17 per dollar, total production costs in one year is calculated as $12 \cdot 1.3 \cdot (1 + 0.17) = 18$. Second, we assume that reaching $1.5 \cdot k$ ($k = 1.3$) within one year incurs twice the capacity adjustment cost required to reach k in the same timeframe. Using these two conditions, we solve for the parameters of the cost function and obtain $a = 1.7, q = 40.8$. We use a standard backward induction algorithm (**?**) to solve for the optimal policy and value. Figure 5.4 presents the resulting growth trajectory. The sales effort shows an overproportional growth over time, suggesting that entrepreneurs should gradually scale their sales efforts. This strategy helps minimize overall costs when capacity adjustment costs are relatively lower—specifically, compared to the linear growth in the quadratic case where $a = 2$. The underlying rationale for this behavior will be further explored in the next section.

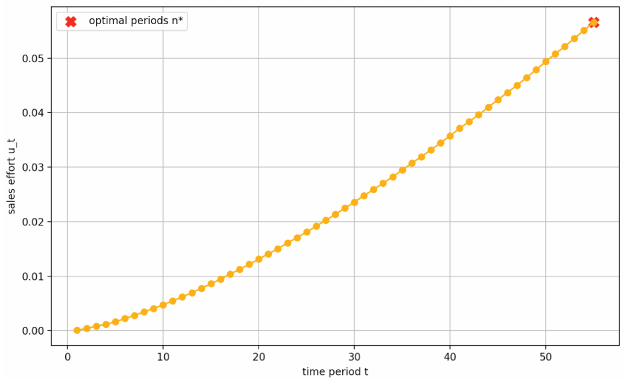


Figure 5.4: Plot of sales effort over time ($a = 1.7$).

5.6.2 Non-Quadratic Capacity Adjustment Cost

Impact of convexity in capacity adjustment costs. Figure 5.5 illustrates the growth trajectories under varying convexity levels in capacity adjustment costs, parameterized by a. When $a < 2$, sales effort increases at a more than proportional rate, with lower values of a leading to a more gradual and sustained growth trajectory. In contrast, when $a = 2$, the effort trajectory follows a linear trend, as expected from Equation (5.16). For $a > 2$, where capacity adjustment costs become highly convex and increasingly penalizing, sales effort u_t grows more aggressively in the early periods but slows down as recurring sales expand.

From Equation (5.5) ($C_n(\bar{u}) = \sum_{t=1}^{n} Q(u_t) - m \cdot \sum_{t=1}^{n} \sum_{i=1}^{t} u_i + c \cdot k$) and Proposition 5.c, which establishes that $u_t^* \leq u_{t+1}^*$, the optimal sales trajectory is shaped by a trade-off: avoiding excessive late-period capacity adjustment costs while leveraging the benefits of late sales effort, driven by the accumulation of costs in production margins: $-m \cdot \sum_{t=1}^{n} \sum_{i=1}^{t} u_i$, which encourages a more backloaded strategy when capacity adjustment costs are less convex. However, when a is big, the convexity of the function $Q(\bar{u}) = q \cdot \bar{u}^a$ increases, meaning increasing u_t becomes more costly at each period, including the final one. This diminishes the incentive to defer sales efforts to later periods. As a result, when a is high, startups have stronger incentives to push more effort early to avoid very costly late-period efforts. Conversely, for lower a, startups can impose more balanced and gradually accelerating effort over time to benefit from late sales effort.

For entrepreneurs, this analysis suggests that when capacity adjustment costs are high ($a > 2$), it is beneficial to adopt an aggressive early growth strategy. In contrast, when adjustment costs are low ($a < 2$), it will be preferable to start steadily with gradual acceleration.

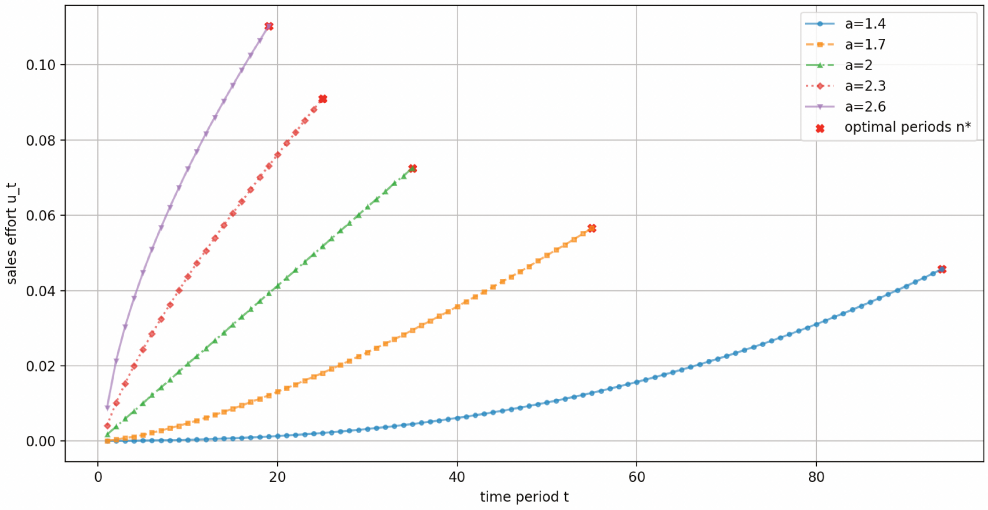


Figure 5.5: Plot of sales effort over time with different convexity levels of capacity adjustment costs.

5.6.3 Stochastic Sales Acquisition

Next, we examine the growth trajectories when new sales W_t from sales efforts u_t are stochastic modeled as $W_t = u_t \cdot Z_t$, where Z_t being the realization at time t of a positive continuous random variable Z with mean μ_t and density function $\varphi(\cdot)$. We use the same parameter values as described in Section 5.6.1, with additional specifications for the stochasticity in new sales. In the base case, Z follows a truncated normal distribution bounded within $[0, 2]$, with the mean of 1 and the variance of 1. For the computational tractability, we discretize the state and sales effort using a step size of 0.01.

Impact of the distribution variance To analyze how distribution variance affects decision making over time, we compare growth trajectories under five variance levels: 0.05, 0.5, 1, 1.5, and 2. Figures 5.6 and 5.7 illustrate the resulting trajectories. Compared to the deterministic setting, uncertainty leads to more cautious investment in sales efforts, particularly by delaying effort increases to later periods. Specifically, as variance increases from 0.05 to 1.5, sales effort is postponed further to later periods. However, beyond a variance of 1.5, the growth trajectories remain nearly identical, possibly due to diminishing sensitivity to variance beyond this threshold. Additionally,

under uncertainty, firms take longer to reach their target and exert lower sales effort in the final periods. These are also reflected in expected accumulated sales over time, where higher variance leads to slower growth and a flatter trajectory in the last periods.

This result is reasonable: with lower variance, sales are more predictable, allowing for steady growth and a more predictable optimal stopping time. In contrast, higher variance introduces greater fluctuations in sales realizations, making the optimal stopping time (i.e., when $x_t = k$ is reached) less predictable. Additionally, since the capacity adjustment cost function is convex and negative production margins accumulate over time, sales effort tend to gradually increase when the firm is far from reaching k. However, once x_t is observed to be close to k, less effort is expected to reach k, leading to a reduction in sales efforts in the final periods.

For entrepreneurs, these findings suggest that in turbulent market characterized by higher demand uncertainty, it is better to grow slowly to see the market situation and then gradually escalate sales effort, and then at final periods, they can reduce sales effort intensity once achieving confidence in meeting the target recurring sales.

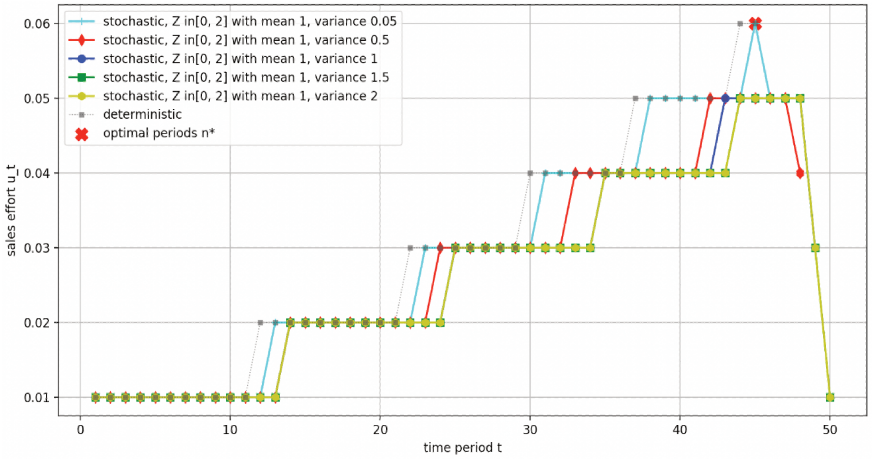


Figure 5.6: Plot of sales effort over time with different variances in the same graph

Impact of convexity in capacity adjustment costs: Figures 5.8 and 5.9 compare growth trajectories under different convexity levels in both deterministic and stochasticity settings. We observe that the stochasticity setting

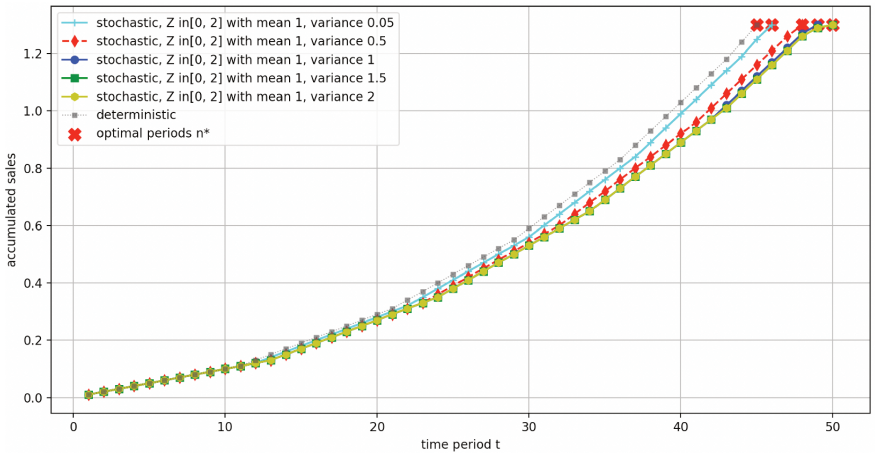


Figure 5.7: Plot of accumulated recurring sales over time with different variances in the same graph

follows a similar pattern to the deterministic case: When convexity level is high (a is larger), sales effort grows more aggressively in the early periods and slows down as capacity expands, especially when far from the target k. As previously discussed, growth under stochasticity is slower than that in the deterministic case. Additionally, we find that the difference in accumulated sales between the two settings is bigger at higher convexity levels, indicating that uncertainty has a stronger impact on growth speed when capacity adjustment costs are highly convex. A possible reason is that under uncertainty, firms hesitate to invest too aggressively early on, as future sales outcomes are unpredictable. As convexity increases, the penalty for large adjustments becomes more severe, making firms in the stochastic setting more cautious. This suggests that in highly volatile markets with significant capacity adjustment costs per unit, entrepreneurs should adopt a more cautious approach to sales growth.

5.7 Conclusions

Unlike established firms that have sufficient time and financial resources to expand market at their own pace, startups seeking to scale up must invest in their own production capabilities under tighter constraints. To secure enough funding for building production facilities, they need a solid customer base. To achieve this, startups should

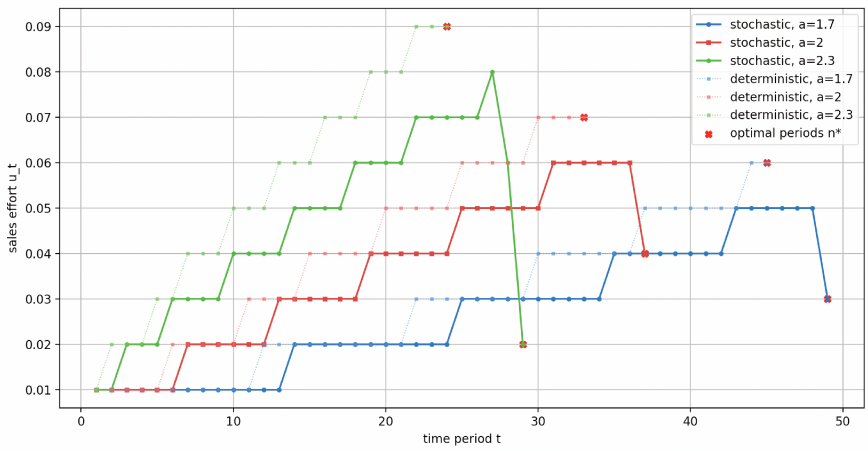


Figure 5.8: Plot of sales effort over time with different convexity levels of capacity adjustment costs

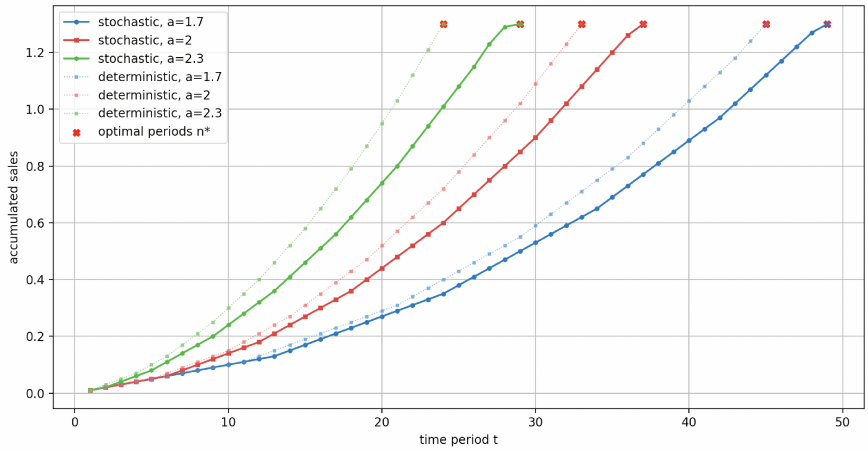


Figure 5.9: Accumulated recurring sales over time with different convexity levels of capacity adjustment costs

balance between period-to-period capacity adjusment costs with the need to reach the target recurring sales within a limited time frame. To address this challenge, we propose a growth model for startups, incorporating capacity adjustment costs, sales effort costs, and sales margins. Using this model, we characterize optimal growth trajectories and assess how factors such as capacity adjustment costs, target recurring sales, and stochasticity in sales acquisition influence startup's growth strategy.

Our results provide entrepreneurs with several key managerial implications. First, instead of pursuing the fastest

possible growth, entrepreneurs should adopt a strategic growth path that balances growth speed with costs. Second, when capacity adjustment costs are high, it is better to grow more aggressively in the early stages and then slow down as recurring sales expand. Third, in a predictable market, entrepreneurs should gradually increase sales efforts over time rather than speed up and down. Fourth, in a highly volatile market environment, a more cautious sales effort strategy is recommended, delaying the increase of sales effort to later periods. Besides, as the startup approaches its target number, less effort is required to minimize total cost.

These insights can be applied to any industry where startups rely on external suppliers to meet demand before achieving full production capabilities. However, our model has some limitations. For example, we asume non-decreasing recurring sales, even in the absence of sales effort imposed. In reality, firms may lose sales without continuous engagement. In the future research, we can extend our model by relaxing the assumption of non-decreasing recurring sales, modeling the dynamic sales effort cost, reflecting the startup's growing market reputation, or integrating production forecasting for CM, which help startups proactively optimize capacity adjustments and reduce associated costs.

6

Conclusion and Future Directions

This dissertation investigates how decision models can support growth-oriented decision-making in two fast-evolving domains: last-mile delivery operations and startup scaling. By addressing challenges related to uncertainty, resource constraints, and timing, it contributes both theoretically and practically to understanding how to guide growth decisions in dynamic environments.

6.1 Summary of contributions

Chapters 2, 3, and 4 study the last-mile delivery operations from complementary angles.

Chapter 2 presents a structured review of emerging optimization problems in same-day delivery logistics. It maps the delivery process, identifies sources of uncertainty, categorizes existing research, and reviews solution approaches. The chapter then compares solution structures derived from different modeling formulations and discusses their practical implications. It also highlights managerial insights and outlines directions for future research.

Chapter 3 addresses the orienteering problem with stochastic and dynamic release dates in same-day delivery. It proposes two reinforcement learning approaches based on value function approximation—one leveraging deterministic integer linear programs with a consensus function and the other incorporating a two-stage stochastic program. By comparing with various benchmark approaches, it reveals the values of incorporating future information, using sampling rather than point estimations, using exact methods rather than heuristics, and using approximations of future routes. This work focuses on the single-vehicle setting and lays a foundation for future exploration of multi-vehicle scenarios, where adapting batch-based approximations remains an open challenge.

Chapter 4 explores a delivery model in which autonomous ground vehicles transport parcels from a central depot to pickup stations via direct, single-destination trips. This chapter focuses on the static and deterministic version of the problem for theoretical analysis and demonstrates that the problem and its variants can be effectively solved with solution approaches adapted from established optimization methods. It also derives performance

bounds and evaluates the proposed heuristics through worst-case and numerical analyses. While the original goal was to study is a more complex version of the problem involving varying parcel release dates, this extension is left for future work due to time constraints and the increased problem difficulty.

Together, these chapters show that integer programming techniques remains highly valuable, even in real-time contexts—especially when combined with reinforcement learning or approximation techniques. This is evident in both Chapters 2 and 3, which highlight the usefulness of integer linear programming models within dynamic and stochastic environments.

At the same time, research attention is increasingly shifting toward stochastic and dynamic last-mile delivery problems. We are also witnessing a surging popularity of AI-related methods, such as deep reinforcement learning approaches, for addressing these complicated optimization problems in real-time.

Chapter 5 shifts the focus to startup scaling. Startups must build a solid customer base to secure funding for production and delivery capabilities. To this end, they need to balance between capacity adjustment costs and the need to reach the target within a limited time frame. Using a dynamic growth model, this chapter illustrates that when adjustment costs are relatively high, entrepreneurs should grow faster in the early stages and then grow slower as they get closer to the target. Under uncertainty, more cautious strategies—delaying increased sales effort—may be more cost-effective. This work helps explain how startups can optimize their growth pace when faced with limited resources and performance risks.

6.2 Limitations

While this dissertation provides useful decision models and managerial insights, several limitations remain.

First, in the last-mile delivery chapters, the models focus mainly on a single vehicle. In Chapter3, while the pro-

posed methods perform well, the computational running time increases with the number of vehicles considered, and adjustments would be needed, such as a tailored batch approach, to maintain efficiency. Additionally, the models consider solely the objective of maximizing parcels delivered with each parcel having the same priority and no time windows. In reality, delivery platforms may need to balance multiple objectives, such as customer priority and environmental impact. Addressing this trade-off explicitly remains a challenging but important direction.

Second, some of the dynamic models assume complete knowledge of model parameters, such as the distribution of demand or travel times. In practice, these may be difficult to estimate or may vary over time. While scenario-based and learning-based methods help with this, more effort is needed to build reliable models that perform well under different sources of uncertainty.

Third, the startup scaling model is assuming a static cost of sales effort. In practice, however, this cost may decrease over time as startup builds brand recognition and market credibility, making it easier and less expensive to attract new sales. An interesting extension would be to analyze the implications of a time-varing or declining sales effort cost on optimal growth trajectories.

Fourth, the current startup work studies only one of the many challenges faced during the scaling phase. In reality, startups face a range of complex issues—including team coordination, evolving customer behavior, market competition, and shifts in product-market fit. The model could be extended to capture these richer dynamics by incorporating additional elements.

6.3 Future research directions

Despite the advanced and continuous development of new delivery technologies and algorithms, achieving fast, cost-effective, and sustainable delivery remains a complex challenge. Rising customer expectations, operational constraints, and growing environmental concerns make it increasingly difficult to optimize for profit and service

quality simultaneously. Future research on last-mile delivery could consider:

- Incorporating environmental impacts into operational models, for example, incentivizing eco-friendly choices among consumers;
- Addressing multi-objective optimization problems that balance profitability, service quality, delivery speed, and driver safety;
- Meanwhile, the research in last-mile delivery is evolving rapidly, and it remains challenging to create impactful research for the long term. It is interesting to explore adaptive strategies for sustainable growth that account for variability in demand, labor, and technology availability;
- There has been promising studies in hybrid approaches that combine the interpretability and guarantees of methods from Operations Research with the adaptability of AI-based algorithms. It remains an exciting area to bridge the methodological gap between Computer Science and Operations Research on solving large-scale optimization problems.

In startup scaling, besides considering specific model elements as specified before, it will also be interesting to investigate more interactions with market response or varying investor expectations into the decision framework.

In both application domains, this dissertation shows that decision models can play a valuable role in guiding growth strategies. By enabling more structured and forward-looking decisions, these models help decision-makers, such as business owners, logistics managers, and entrepreneurs, handle uncertainty, avoid making myopic decisions, and pursue more stable and effective growth strategies.

While this dissertation focuses on last-mile delivery problems faced by established companies aiming for sustainable growth, it is worth noting that many last-mile delivery startups have struggled or failed in the market. A promising future direction would be to bridge the two application domains more explicitly by studying the unique

challenges faced by such startups and investigating relevant strategies, such as a recent paper that considers the evolution of customer demand for on-demand, same-day delivery startups (Chen et al., 2025).

As the fields of logistics and entrepreneurship continue evolving, it is increasingly important to integrate operations research methodologies with operations management perspectives to develop effective and sustainable growth models.

A

Appendix

A.1 Appendix-Chapter 4

This appendix contains the proofs of the results for Chapter 5.

Proof. **Proof of Proposition 5.**

Part (a): The total cost C_n can be minimized sequentially, first over u_1, then over $u_2, \ldots, u_{n-1}$. The function can be written as:

$$\min_{\bar{u}} C_n(\bar{u}) = \min_{u_1}(f_1(u_1) + g^*(u_1)), \tag{A.1}$$

where $f_1(u_1)$ represents the cost of the first period, and $g^*(u_1)$ is the optimal cost of the remaining $n-1$ periods, conditioned on the accumulated u_1 sales.

Since $Q(\cdot)$ is convex and the other costs of $f_1(u_1)$ are linear, $f_1(u_1)$ is convex in u_1. Moreover, minimizing a convex function over some variables preserves the convexity (see P144 in Boyd & Vandenberghe (2004)), so $g^*(u_1)$ is convex in u_1. Thus the sum of the two convex functions is convex in u_1. After adding the non-negativity constraint, it is still convex. Following the convexity, if $u_1^* < 0$ in the unconstrained problem, then the non-negativity constraint $u_1 \geq 0$ will bind. This implies that the constrained optimal solution will be $\hat{u}_1^* = 0$. The same argument applies to any $t \in \{1, \ldots, n-1\}$. The cost function can be seen as:

$$C_n(\bar{u}) = f_t(u_t) + g^*(u_t), \tag{A.2}$$

which proves that $u_t^* < 0$ implies $\hat{u}_t^* = 0$. where $f_t(u_t)$ is the cost of period t, and $g^*(u_t)$ is the cost of the remaining $n-1$ periods given u_t. Both functions are convex, so $C_n(\bar{u})$ is convex in u_t. Therefore, if $u_t^* < 0$ in the unconstrained problem, then the non-negativity constraint $u_t \geq 0$ will bind, and $\hat{u}_t^* = 0$ in the constrained

problem.

Part (b) follows from the fact that if $u_1 = 0$, the optimization function of (5.6) with n periods as well as the constraints reduce to the optimization problem with $n-1$ periods, which can be easily observed from (5.5).

Part (c) follows from the fact that for $t < n-1$, the derivative $(\partial C_n / \partial u_{t+1})$ at u_t^* is negative (because the term $m(n-t-1)$ is greater than $m(n-t)$ with $m < 0$), such that u_{t+1} has to be greater than u_t^* in order to have its derivative taking a value of zero. When $t = n-1$, as the derivative (see (5.7)) we have $Q'(u_{n-1}) - Q'(k - \sum_{i=1}^{n-1} u_i) - m = Q'(u_{n-1}) - Q'(u_n) - m = 0$. Since $m < 0$ and $Q(\cdot)$ is convex-increasing, $u_{n-1} < u_n$.

For Part (d), when $u_1^*|n-1 < 0$, from the optimization problem with $n-1$ periods to the problem with n periods, there can be three possible cases:

- 1) $u_1^* \leq 0$, which matches the proposition;
- 2) $u_1^* = a > 0, u_2^* \leq 0$, which cannot happen because of part (c) of Proposition (5);
- 3) $u_1^* = a > 0, u_2^* = b > 0$. We now show that this case cannot occur.

  For the (n-1)-period problem, we can rewrite the unconstrained (without $u_t \geq 0$) optimization problem of (5.6) with Lagrange multiplier λ for the constraints $\sum_{t=1}^{n-1} u_t = k$:

$$\mathcal{L}_{n-1}(\lambda, \bar{u}) = \sum_{t=1}^{n-1} Q(u_t) - m \cdot \sum_{t=1}^{n-1} \sum_{i=1}^{t} u_i + c \cdot \sum_{t=1}^{n-1} u_t + \lambda(\sum_{t=1}^{n-1} u_t - k), \tag{A.3}$$

  For the new n-period problem, assuming $u_1^* = a > 0$, we have: $\hat{u}_2^*|n + \sum_3^n \hat{u}_t^*|n = k - a$. Applying part (b) of this proposition, we translate this into the (n-1)-period problem: $\hat{u}_1^*|(n-1) + \sum_2^n \hat{u}_t|(n-1) = k - a$.

This gives a new Lagrange function:

$$\mathcal{L}_{n-1}(\lambda, \bar{u}) = \sum_{t=1}^{n-1} Q(u_t) - m \cdot \sum_{t=1}^{n-1} \sum_{i=1}^{t} u_i + c \cdot \sum_{t=1}^{n-1} u_t + \lambda(\sum_{t=1}^{n-1} u_t - k + a), \tag{A.4}$$

From the first-order condition of (A.4), we obtain the following system that characterizes the optimal solution:

$$\frac{\partial \mathcal{L}_{n-1}}{\partial \lambda} = 0 \Leftrightarrow \sum_{t=1}^{n-1} u_t - k + a = 0 \tag{A.5}$$

$$\begin{aligned} \frac{\partial \mathcal{L}_{n-1}}{\partial u_t} = 0 &\Leftrightarrow Q'(u_t) - m(n-t) + c + \lambda = 0 \\ &\Leftrightarrow Q'(u_t) - m(n-t) + c = -\lambda \end{aligned} \tag{A.6}$$

Since the constraint is now tighter ($k - a$ instead of k), the optimizer must reduce $\sum_{t=1}^{n-1} u_t$ to satisfy (A.5). Moreover, from (A.6), the change in each $u_t, t \in 1, \ldots, n-1$ must follow the same direction, i.e., all reduce.

Thus, it cannot happen that $u_1^* = a > 0$ and $u_2^* = b > 0$ in n-period problem, given that $u_1^* < 0$ in the (n-1)-period problem. This contradiction completes the proof.

□

Proof. **Proof of Theorem 3.** Similar to the discussion of the proof of part (b) of Proposition 5, the optimal solution of the n-period problem with one period delay $(0, \hat{u}^*|n)$ is a feasible solution but does not necessarily guarantee optimal solution for the $(n+1)$-period problem. In other words, we can always just set sales effort to zero in the first period of an $(n+1)$-period problem and we get exactly the same costs as an n-period problem but that starts one period later (if $x_0 = 0$). Now, when we further minimize the costs of the $(n+1)$-period problem, they can only be lower than or equal to the total costs of the solution of the n-period problem with a leading zero. It follows that

$$C_{n+1}(\hat{u}^*|n+1) \leq C_{n+1}(0, \hat{u}^*|n) = C_n(\hat{u}^*), \tag{A.7}$$

i.e., total minimal costs are decreasing (non-strictly) in n. Hence, the largest n for which costs are still decreasing is optimal.

Further, from parts (a) and (b) of Proposition 5, we know that if $u_1^* < 0$ for a value of n, then the optimal constrained solution is $\hat{u}_1^* = 0$ and the remaining values $\hat{u}_t^*$ with $t > 1$ are identical to the optimal values of the problem with $n-1$ periods (and consequently the total costs are also identical). That means that, if $u_1^* \leq 0$ for a problem with n periods, the minimum total costs are identical to those of the problem with $n-1$ periods. Also, part (d) of Proposition 5 implies that the same will hold for any other value $n' > n$. Hence, total costs do not decrease anymore after the greatest n for which $u_1^* > 0$.

Both of the above results imply that the greatest period less than n for which $u_1^* > 0$ is (one of the) optimal period values, and we label it n^*.

Finally, for the optimal values of sales efforts, note that for a problem with n^* periods, $u_t^* \geq u_1^*$ for $t = 1...n^*$ from successively applying part (c) of Proposition 5, and $u_1^* > 0$ by definition of our n^*. Hence, all unconstrained optimal values given by (5.7) are non-negative. Thus, the unconstrained optimal values are also optimal for the constrained problem (5.6). □

Proof. **Proof of Proposition 6.** With expression (5.15), we assume that $n' = \frac{1}{2} + \frac{1}{2}\sqrt{\frac{m-16kq}{m}}$. Let's take the following derivative:

$$\begin{aligned} \frac{dn'}{dk} &= \frac{d}{dk}\left(\frac{1}{2} + \frac{1}{2}\sqrt{\frac{m-16kq}{m}}\right) \\ &= -\frac{4q}{m\sqrt{\frac{m-16kq}{m}}} > 0, \forall t = 1...n-1, \end{aligned} \tag{A.8}$$

Thus when k increases, n' increases. From (5.15), we know that the optimal number of periods n^* will be non-decreasing in k. □

Proof. **Proof of Proposition 7.** We get the first derivative of the expression (5.15) with respect to m:

$$\frac{d}{dm}(\frac{1}{2}+\frac{1}{2}\sqrt{\frac{m-16kq}{m}}) = \frac{4kq}{m^2\sqrt{\frac{m-16kq}{m}}} > 0,$$

This derivative shows how the expression changes with respect to m. Given that $m < 0$, the positive derivative indicates that increasing m (or equivalently, decreasing $|m|$) leads to an increase in the expression. Conversely, if $|m|$ increases (i.e., m becomes more negative), the expression decreases. Therefore, n^* decreases as $|m|$ increases. □

Proof. **Proof of Proposition 8.** In the case of a quadratic capacity adjustment cost $Q(u) = qu^2$, with $q > 0$, the startup's cost optimization problem in period N becomes:

$$\min_{u_N} g_N = qu_N^2 + c \cdot u_N - m \cdot (x_N + u_N) + p(k - x_N - u_N), \text{ subject to } 0 \leq u_N \leq k - x_N. \tag{A.9}$$

Solving the unconstrained version yields the optimal sales effort u_N^*:

$$u_N^* = \frac{m - c + p}{2q}. \tag{A.10}$$

With the linear constraints added, the problem is still convex, $\hat{u}_N^* = min([\frac{m-c+p}{2q}]^+, k - x_N)$. From the previous analysis (see (5.11)), we know $x_N = \sum_{t=1}^{N-1} u_t^* = \frac{(N-1)(mN+4qk)}{4qN}$. We have

$$\hat{u}_N^* = \begin{cases} \frac{m-c+p}{2q}, & \text{if } \frac{(N-1)(mN+4qk)}{4qN} \leq k - \frac{m-c+p}{2q} \\ k - \frac{(N-1)(mN+4qk)}{4qN}, & \text{otherwise.} \end{cases}$$

□

References

Ackva, C. & Ulmer, M. (2024). Consistent Routing for Local Same-Day Delivery via Micro-Hubs. *OR Spectrum*, 46(2), 375–409.

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.

Agatz, N., Fan, Y., & Stam, D. (2021). The impact of green labels on time slot choice and operational sustainability. *Production and Operations Management*, 30(7), 2285–2303.

Alfandari, L., Ljubić, I., & da Silva, M. D. M. (2022). A tailored benders decomposition approach for last-mile delivery with autonomous robots. *European Journal of Operational Research*, 299(2), 510–525.

Alumur, S. A., Campbell, J. F., Contreras, I., Kara, B. Y., Marianov, V., & O'Kelly, M. E. (2021). Perspectives on modeling hub location problems. *European Journal of Operational Research*, 291(1), 1–17.

Angelelli, E., Archetti, C., Filippi, C., & Vindigni, M. (2021). A dynamic and probabilistic orienteering problem. *Computers & Operations Research*, 136, 105454.

Anuar, W. K., Lee, L. S., Seow, H.-V., & Pickl, S. (2021). A multi-depot vehicle routing problem with stochastic road capacity and reduced two-stage stochastic integer linear programming models for rollout algorithm. *Mathematics*, 9(13), 1572.

Archetti, C. & Bertazzi, L. (2021). Recent challenges in routing and inventory routing: E-commerce and last-mile delivery. *Networks*, 77(2), 255–268.

Archetti, C., Feillet, D., Mor, A., & Speranza, M. G. (2018). An iterated local search for the traveling salesman problem with release dates and completion time minimization. *Computers & Operations Research*, 98, 24–37.

Archetti, C., Feillet, D., Mor, A., & Speranza, M. G. (2020a). Dynamic traveling salesman problem with stochastic release dates. *European Journal of Operational Research*, 280(3), 832–844.

Archetti, C., Feillet, D., Mor, A., & Speranza, M. G. (2020b). Problem instances of reinforcement learning for orienteering problem with stochastic and dynamic release dates. https://drive.google.com/file/d/1mVw2u1vuvq6NooKbfHlcYPQReBJXqimQ/view. Accessed: 2023-07-14.

Archetti, C., Feillet, D., & Speranza, M. G. (2015). Complexity of routing problems with release dates. *European journal of operational research*, 247(3), 797–803.

Archetti, C., Guerriero, F., & Macrina, G. (2021). The online vehicle routing problem with occasional drivers. *Computers & Operations Research*, 127, 105144.

Archetti, C., Savelsbergh, M., & Speranza, M. G. (2016). The vehicle routing problem with occasional drivers. *European Journal of Operational Research*, 254(2), 472–480.

Archetti, C., Speranza, M. G., & Vigo, D. (2014). Chapter 10: Vehicle routing problems with profits. In *Vehicle routing: Problems, methods, and applications, second edition* (pp. 273–297). SIAM.

Arruñada, B. & Vázquez, X. H. (2006). When your contract manufacturer becomes your competitor. *Harvard business review*, 84(9), 135.

Arslan, A. M., Agatz, N., & Klapp, M. A. (2021). Operational strategies for on-demand personal shopper services. *Transportation Research Part C: Emerging Technologies*, 130, 103320.

Arslan, A. M., Agatz, N., Kroon, L., & Zuidwijk, R. (2018). Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers. *Transportation Science*, 53(1), 222–235.

Auad, R., Erera, A., & Savelsbergh, M. (2023). Courier satisfaction in rapid delivery systems using dynamic operating regions. *Omega*, 121(4), 102917.

Ausseil, R., Pazour, J. A., & Ulmer, M. W. (2022). Supplier menus for dynamic matching in peer-to-peer transportation platforms. *Transportation Science*, 56(5), 1304–1326.

Azi, N., Gendreau, M., & Potvin, J.-Y. (2012). A dynamic vehicle routing problem with multiple delivery routes. *Annals of Operations Research*, 199(1), 103–112.

Babich, V. & Sobel, M. J. (2004). Pre-IPO operational and financial decisions. *Management Science*, 50(7), 935–948.

Bakach, I., Campbell, A. M., & Ehmke, J. F. (2022). Robot-based last-mile deliveries with pedestrian zones. *Frontiers in Future Transportation*, 2.

Banerjee, D., Erera, A., & Toriello, A. (2024). Pricing and demand management for integrated same-day and next-day delivery systems. *Transportation Science*, 0(0).

Banerjee, D., Erera, A. L., Stroh, A. M., & Toriello, A. (2023). Who has access to e-commerce and when? time-varying service regions in same-day delivery. *Transportation Research Part B: Methodological*, 170, 148–168.

Banerjee, D., Erera, A. L., & Toriello, A. (2022). Fleet sizing and service region partitioning for same-day delivery systems. *Transportation Science*, 56(5), 1327–1347.

Barbee, J., Jayakumar, J., Touse, S., , & Venkataraman, K. (2021). Retail's need for speed: Unlocking value in omnichannel delivery. https://www.mckinsey.com/industries/retail/our-insights/retails-need-for-speed-unlocking-value-in-omnichannel-delivery. Accessed: 2022-07-26.

Barr, A. & Wohl, J. (2013). Exclusive: Wal-mart may get customers to deliver packages to online buyers. https://www.reuters.com/article/us-retail-walmart-delivery-idUSBRE92R03820130328. Accessed: 2022-11-01.

Basso, R., Kulcsár, B., Sanchez-Diaz, I., & Qu, X. (2022). Dynamic stochastic electric vehicle routing with safe reinforcement learning. *Transportation research part E: logistics and transportation review*, 157, 102496.

Behrend, M. & Meisel, F. (2018). The integration of item-sharing and crowdshipping: Can collaborative consumption be pushed by delivering through the crowd? *Transportation Research Part B: Methodological*, 111, 227–243.

Belieres, S., Hewitt, M., Jozefowiez, N., & Semet, F. (2021). A time-expanded network reduction matheuristic for the logistics service network design problem. *Transportation Research Part E: Logistics and Transportation Review*, 147, 102203.

Bensinger, G. (2015). Amazon's next delivery drone: You. https://www.wsj.com/articles/amazon-seeks-help-with-deliveries-1434466857. Accessed: 2022-11-01.

Bensoussan, A. & Chevalier-Roignant, B. (2019). Sequential capacity expansion options. *Operations research*, 67(1), 33–57.

Bent, R. W. & Van Hentenryck, P. (2004). Scenario-based planning for partially dynamic vehicle routing with stochastic customers. *Operations Research*, 52(6), 977–987.

Bertsekas, D. P. (2019). *Reinforcement learning and optimal control*. Athena Scientific.

Bertsekas, D. P., Tsitsiklis, J. N., & Wu, C. (1997). Rollout algorithms for combinatorial optimization. *Journal of Heuristics*, 3, 245–262.

Bertsimas, D. J. & Van Ryzin, G. (1991). A stochastic and dynamic vehicle routing problem in the Euclidean plane. *Operations Research*, 39(4), 601–615.

Beyond Meat, Inc. (2021). Beyond meat 2020 annual report. Accessed: 2025-03-31.

Beyond Meat, Inc. (2024). Beyond meat® reports fourth quarter and full year 2023 financial results. Accessed: 2025-03-31.

Bhaskaran, S., Erzurumlu, S. S., & Ramachandran, K. (2021). Sequential product development and introduction by cash-constrained start-ups. *Manufacturing & Service Operations Management*, 23(6), 1505–1523.

Birge, J. R. & Louveaux, F. (2011). *Introduction to stochastic programming*. Springer Science & Business Media.

Bock, S. (2021). Pro-active strategies in online routing. *Supply Chain Management in Manufacturing and Service Systems: Advanced Analytics for Smarter Decisions*, 304, 205–239.

Boyd, S. P. & Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Boysen, N., Fedtke, S., & Schwerdfeger, S. (2021). Last-mile delivery concepts: a survey from an operational research perspective. *Or Spectrum*, 43(1), 1–58.

Bozanta, A., Cevik, M., Kavaklioglu, C., Kavuk, E. M., Tosun, A., Sonuc, S. B., Duranel, A., & Basar, A. (2022). Courier routing and assignment for food delivery service using reinforcement learning. *Computers & Industrial Engineering*, 164, 107871.

Burk, D. C. & Averbeck, B. B. (2023). Environmental uncertainty and the advantage of impulsive choice strategies. *PLOS Computational Biology*, 19(1), e1010873.

Businesscoot (2022). Umiami – profil d'entreprise. Accessed: 2025-03-31.

Cattaruzza, D., Absi, N., & Feillet, D. (2016). The multi-trip vehicle routing problem with time windows and release dates. *Transportation Science*, 50(2), 676–693.

Cavallo, A., Ghezzi, A., Dell'Era, C., & Pellizzoni, E. (2019). Fostering digital entrepreneurship from startup to scaleup: The role of venture capital funds and angel groups. *Technological Forecasting and Social Change*, 145, 24–35.

Center, S. O. (2022). The worst mile: Production pressure and the indury crisis in Amazon's delivery system. Accessed: 2024-11-07.

Cerulli, M., Archetti, C., Fernandez, E., & Ljubic, I. (2024). A bilevel approach for compensation and routing decisions in last-mile delivery. *Transportation Science*, 58(5), 1076–1100.

Chand, S., Hsu, V. N., & Sethi, S. (2002). Forecast, solution, and rolling horizons in operations management problems: A classified bibliography. *Manufacturing & Service Operations Management*, 4(1), 25–43.

Chao, X., Chen, H., & Zheng, S. (2009). Dynamic capacity expansion for a service firm with capacity deterioration and supply uncertainty. *Operations research*, 57(1), 82–93.

Chen, X., Ulmer, M. W., & Thomas, B. W. (2022). Deep Q-learning for same-day delivery with vehicles and drones. *European Journal of Operational Research*, 298(3), 939–952.

Chen, X., Ulmer, M. W., & Thomas, B. W. (2025). To start up a start-up — embedding strategic demand development in operational on-demand fulfillment via reinforcement learning with information shaping. *arXiv preprint arXiv:2504.05633*.

Chen, X., Wang, T., Thomas, B. W., & Ulmer, M. W. (2023). Same-day delivery with fair customer service. *European Journal of Operational Research*, 308(2), 738–751.

Chung, S. H., Sah, B., & Lee, J. (2020). Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions. *Computers & Operations Research*, 123, 105004.

Churchill, N. C. & Mullins, J. W. (2001). How fast can your company afford to grow. *Harvard business review*, 79(5), 135–143.

Cohen, R. & Katzir, L. (2008). The generalized maximum coverage problem. *Information Processing Letters*, 108(1), 15–22.

Côté, J.-F., de Queiroz, T. A., Gallesi, F., & Iori, M. (2023). A branch-and-regret algorithm for the same-day delivery problem. *Transportation Research Part E: Logistics and Transportation Review*, 177, 103226.

Crainic, T. G., Maggioni, F., Perboli, G., & Rei, W. (2018). Reduced cost-based variable fixing in two-stage stochastic programming. *Annals of Operations Research*, (pp. 1–37).

Crowdcube (2018). Daily dose pitch page. Accessed: 2025-03-31.

Cui, R., Li, M., & Li, Q. (2020). Value of high-quality logistics: Evidence from a clash between SF Express and Alibaba. *Management Science*, 66(9), 3879–3902.

Daganzo, C. F. (1984). The length of tours in zones of different shapes. *Transportation Research Part B: Methodological*, 18(2), 135–145.

Dai, H. & Liu, P. (2020). Workforce planning for O2O delivery systems with crowdsourced drivers. *Annals of Operations Research*, 291(1), 219–245.

Davis, E. (2021). The work ahead: Transportation and logistics delivering on the digital- physical value chain. https://www.cognizant.com/us/en/whitepapers/documents/the-work-ahead-transportation-and-logistics-delivering-on-the-digital-physical-value-chain-codex6763.pdf. Accessed: 2022-07-19.

Dayarian, I. & Savelsbergh, M. (2020). Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders. *Production and Operations Management*, 29(9), 2153–2174.

Dayarian, I., Savelsbergh, M., & Clarke, J.-P. (2020). Same-day delivery with drone resupply. *Transportation Science*, 54(1), 229–249.

Dell'Amico, M., Montemanni, R., & Novellani, S. (2023). Pickup and delivery with lockers. *Transportation Research Part C: Emerging Technologies*, 148, 104022.

DHL (2024). The real value of IoT in supply chains. Accessed: 2024-11-02.

Digital Food Lab (2024). Beyond meat & impossible foods still growing fast, with lower margins. Accessed: 2025-03-31.

Dobush, G. (2024). 'Amazon Day' shipping option aims to reduce deliveries and co2 emissions. Accessed: 2024-11-02.

Dragomir, A. G., Van Woensel, T., & Doerner, K. F. (2022). The pickup and delivery problem with alternative locations and overlapping time windows. *Computers & Operations Research*, 143, 105758.

Dumez, D., Lehuédé, F., & Péton, O. (2021). A large neighborhood search approach to the vehicle routing problem with delivery options. *Transportation Research Part B: Methodological*, 144, 103–132.

Dupačová, J., Consigli, G., & Wallace, S. W. (2000). Scenarios for multistage stochastic programs. *Annals of operations research*, 100, 25–53.

Eisenmann, T. (2021). Why startups fail: It's not always the horse or the jockey. *Harvard Business Review*, 99(3), 76.

Ferrucci, F. & Bock, S. (2016). Pro-active real-time routing in applications with multiple request patterns. *European Journal of Operational Research*, 253(2), 356–371.

Ferrucci, F., Bock, S., & Gendreau, M. (2013). A pro-active real-time control approach for dynamic vehicle routing problems dealing with the delivery of urgent goods. *European Journal of Operational Research*, 225(1), 130–141.

Fine, C. H., Padurean, L., & Naumov, S. (2022). Operations for entrepreneurs: Can operations management make a difference in entrepreneurial theory and practice? *Production and Operations Management*.

Fischetti, M., Salazar González, J. J., & Toth, P. (1998). Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing*, 10, 133–148.

Fleckenstein, D., Klein, R., & Steinhardt, C. (2023). Recent advances in integrating demand management and vehicle routing: A methodological review. *European Journal of Operational Research*, 306(2), 499–518.

Fu, J. & Ma, L. (2022). Long-haul vehicle routing and scheduling with biomathematical fatigue constraints. *Transportation Science*, 56(2), 404–435.

Galiullina, A., Mutlu, N., Kinable, J., & Van Woensel, T. (2024). Demand steering in a last-mile delivery problem with home and pickup point delivery options. *Transportation Science*, 58(2), 454–473.

Garcia, A. & Shen, Z. (2010). Equilibrium capacity expansion under stochastic demand growth. *Operations research*, 58(1), 30–42.

Ghiani, G., Manni, A., & Manni, E. (2022). A scalable anticipatory policy for the dynamic pickup and delivery problem. *Computers & Operations Research*, 147, 105943.

Ghiani, G., Manni, E., Quaranta, A., & Triki, C. (2009). Anticipatory algorithms for same-day courier dispatching. *Transportation Research Part E: Logistics and Transportation Review*, 45(1), 96–106.

Gianfelice, M., Aboshosha, H., & Ghazal, T. (2022). Real-time wind predictions for safe drone flights in Toronto. *Results in Engineering*, 15, 100534.

Gleim, M. R., Smith, J. S., Andrews, D., & Cronin Jr, J. J. (2013). Against the green: A multi-method examination of the barriers to green consumption. *Journal of retailing*, 89(1), 44–61.

GNC (2024). GNC and Uber team up to deliver health and wellness products directly to your door. Accessed: 2024-11-02.

Goldberg, A. V. & Tarjan, R. E. (1988). A new approach to the maximum-flow problem. *Journal of the ACM (JACM)*, 35(4), 921–940.

Golden, B., Assad, A., Levy, L., & Gheysens, F. (1984). The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11(1), 49–66.

Goodson, J. C., Ohlmann, J. W., & Thomas, B. W. (2013). Rollout policies for dynamic solutions to the multi-vehicle routing problem with stochastic demand and duration limits. *Operations Research*, 61(1), 138–154.

Gray, J. V., Tomlin, B., & Roth, A. V. (2009). Outsourcing to a powerful contract manufacturer: The effect of learning-by-doing. *Production and Operations Management*, 18(5), 487–505.

Gunawan, A., Lau, H. C., & Vansteenwegen, P. (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2), 315–332.

Hamermesh, D. S. & Pfann, G. A. (1996). Adjustment costs in factor demand. *Journal of Economic literature*, 34(3), 1264–1292.

Haugland, D., Ho, S. C., & Laporte, G. (2007). Designing delivery districts for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 180(3), 997–1010.

Hildebrandt, F. D., Thomas, B. W., & Ulmer, M. W. (2023). Opportunities for reinforcement learning in stochastic dynamic vehicle routing. *Computers & Operations Research*, 150, 106071.

Hildebrandt, F. D. & Ulmer, M. W. (2021). Supervised learning for arrival time estimations in restaurant meal delivery. *Transportation Science*, 56(4), 1058–1084.

Hou, J. & Shen, H. (2022). Contract manufacturer's market entry and its impact under asymmetric customer loyalty. *Journal of Systems Science and Systems Engineering*, 31(2), 174–203.

Howland, D. (2016). Amazon pledges same-day delivery to all urban neighborhoods after outcry. https://www.retaildive.com/news/amazon-pledges-same-day-delivery-to-all-urban-neighborhoods-after-outcry/418843/. Accessed: 2022-07-16.

Hu, Q., Kouvelis, P., Xiao, G., & Guo, X. (2022). Horizontal outsourcing and price competition: The role of sole sourcing commitment. *Production and Operations Management*, 31(8), 3198–3216.

Isenberg, D. (2012). Focus entrepreneurship policy on scale-up, not start-up. *Harvard Business Review*, 21(2012), 2–4.

Jabali, O., Gendreau, M., & Laporte, G. (2012). A continuous approximation model for the fleet composition problem. *Transportation Research Part B: Methodological*, 46(10), 1591–1606.

Joerss, M., Neuhaus, F., & Schröder, J. (2016). How customer demands are reshaping last-mile delivery. *The McKinsey Quarterly*, 17, 1–5.

Joglekar, N. R. & Lévesque, M. (2009). Marketing, r&d, and startup valuation. *IEEE Transactions on Engineering Management*, 56(2), 229–242.

Joseph, J. (2025). Driverless utility vehicles are taking over boring and risky jobs. https://www.earth.com/news/driverless-utility-vehicles-are-taking-over-boring-and-risky-jobs/. Accessed: 2025-02-06.

Jules, C., Kshirsagar, A., & Lloyd-George, K. (2022). *Scaling up: How founder CEOs and teams can go beyond aspiration to ascent*. Technical report, Technical Report. McKinsey & Company.

Kagan, E., Hyndman, K. B., & Qi, A. (2022). Venture deals: Sources of leverage in entrepreneur-investor bargaining. *Johns Hopkins Carey Business School Research Paper*, (21-04).

Kahr, M. (2022). Determining locations and layouts for parcel lockers to support supply chain viability at the last mile. *Omega*, 113, 102721.

Kellerer, H., Pferschy, U., Pisinger, D., Kellerer, H., Pferschy, U., & Pisinger, D. (2011). *Knapsack Problems*. Springer.

Khuller, S., Moss, A., & Naor, J. S. (1999). The budgeted maximum coverage problem. *Information processing letters*, 70(1), 39–45.

Klapp, M. A., Erera, A. L., & Toriello, A. (2018a). The dynamic dispatch waves problem for same-day delivery. *European Journal of Operational Research*, 271(2), 519–534.

Klapp, M. A., Erera, A. L., & Toriello, A. (2018b). The one-dimensional dynamic dispatch waves problem. *Transportation Science*, 52(2), 402–415.

Klapp, M. A., Erera, A. L., & Toriello, A. (2020). Request acceptance in same-day delivery. *Transportation Research Part E: Logistics and Transportation Review*, 143, 102083.

Klein, V. & Steinhardt, C. (2023). Dynamic demand management and online tour planning for same-day delivery. *European Journal of Operational Research*, 307, 860–886.

Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2016). Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research*, 249(1), 1–21.

Kundu, A., Anderson, S. J., & Ramdas, K. (2024). Disruptions, redundancy strategies, and performance of small firms: Evidence from uganda. *Management Science*.

Lan, L., van Doorn, J. M., Wouda, N. A., Rijal, A., & Bhulai, S. (2024). An iterative sample scenario approach for the dynamic dispatch waves problem. *Transportation Science*, 58(4), 726–740.

Laporte, G., Asef-Vaziri, A., & Sriskandarajah, C. (1996). Some applications of the generalized travelling salesman problem. *Journal of the Operational Research Society*, 47(12), 1461–1467.

Leung, E. K., Ouyang, Z., & Huang, G. Q. (2023). Community logistics: a dynamic strategy for facilitating immediate parcel delivery to smart lockers. *International Journal of Production Research*, 61(9), 2937–2962.

Lévesque, M. & Stephan, U. (2020). It's time we talk about time in entrepreneurship. *Entrepreneurship Theory and Practice*, 44(2), 163–184.

Lévesque, M., Zhao, X., & Bian, J. (2017). Competitive interplay of production decisions: rivalry between established and startup firms. *IEEE Transactions on Engineering Management*, 65(1), 85–98.

Li, Y., Archetti, C., & Ljubic, I. (2024a). Emerging optimization problems for distribution in same-day delivery. *arXiv preprint arXiv:2405.05620*.

Li, Y., Archetti, C., & Ljubić, I. (2024b). Reinforcement learning approaches for the orienteering problem with stochastic and dynamic release dates. *Transportation Science*, 58(5), 1143–1165.

Li, Y., Archetti, C., & Ljubić, I. (2023). Data of 100 customers of reinforcement learning for orienteering problem with stochastic and dynamic release dates. https://github.com/Yuanyuan517/RL-for-OP-with-stochastic-and-dynamic-RD.git. Accessed: 2024-01-09.

Lim, K. & Ong, J. (2022). The big read: Faced with fatigue and rush to meet orders, food delivery riders grapple daily with road safety risks. https://www.channelnewsasia.com/singapore/food-delivery-riders-road-accidents-safety-rush-orders-big-read-2815206. Accessed: 2022-11-06.

Lin, J., Zhou, W., & Du, L. (2018). Is on-demand same day package delivery service green? *Transportation Research Part D: Transport and Environment*, 61, 118–139.

Liu, S., He, L., & Max Shen, Z.-J. (2021). On-time last-mile delivery: Order assignment with travel-time predictors. *Management Science*, 67(7), 4095–4119.

Liu, W., Liu, L., & Qi, X. (2024). Drone resupply with multiple trucks and drones for on-time delivery along given truck routes. *European Journal of Operational Research*.

Ljubić, I. (2021). Solving Steiner trees: Recent advances, challenges, and perspectives. *Networks*, 77(2), 177–204.

Lu, L. & Yan, X. (2016). Capacity investment decisions under risk aversion. *Naval Research Logistics (NRL)*, 63(3), 218–235.

Lucena, A. & Resende, M. G. (2004). Strong lower bounds for the prize collecting Steiner problem in graphs. *Discrete Applied Mathematics*, 141(1-3), 277–294.

Luo, Z., Chen, X., & Wang, X. (2024). Exploring optimal outsourcing strategy with and without transfer payment. *Naval Research Logistics (NRL)*.

MacQueen, J. (1967). Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability* (pp. 281–297).: University of California Los Angeles LA USA.

Macrina, G., Pugliese, L. D. P., Guerriero, F., & Laporte, G. (2020). Drone-aided routing: A literature review. *Transportation Research Part C: Emerging Technologies*, 120, 102762.

Mancini, S. & Gansterer, M. (2021). Vehicle routing with private and shared delivery locations. *Computers & Operations Research*, 133, 105361.

Mao, C. & Shen, Z. (2018). A reinforcement learning framework for the adaptive routing problem in stochastic time-dependent network. *Transportation Research Part C: Emerging Technologies*, 93, 179–197.

Mao, W., Ming, L., Rong, Y., Tang, C. S., & Zheng, H. (2019). Faster deliveries and smarter order assignments for an on-demand meal delivery platform. *Available at SSRN 3469015*.

Mitrović-Minić, S. & Laporte, G. (2004). Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B: Methodological*, 38(7), 635–655.

Mohri, S. S., Ghaderi, H., Van Woensel, T., Mohammadi, M., Nassir, N., & Thompson, R. G. (2024). Contextualizing alternative delivery points in last mile delivery. *Transportation Research Part E: Logistics and Transportation Review*, 192, 103787.

Moshref-Javadi, M. & Winkenbach, M. (2021). Applications and research avenues for drone-based models in logistics: A classification and review. *Expert Systems with Applications*, 177, 114854.

Muñoz-Villamizar, A., Velázquez-Martínez, J. C., Haro, P., Ferrer, A., & Mariño, R. (2021). The environmental impact of fast shipping ecommerce in inbound logistics operations: A case study in Mexico. *Journal of Cleaner Production*, 283, 125400.

Neira, D. A., Aguayo, M. M., De la Fuente, R., & Klapp, M. A. (2020). New compact integer programming formulations for the multi-trip vehicle routing problem with time windows. *Computers & Industrial Engineering*, 144, 106399.

Neria, G., Hildebrandt, F. D., Tzur, M., & Ulmer, M. W. (2024). The restaurant meal delivery problem with ghost kitchens. *Transportation Science*.

Nesin, B. (2022). Will customer stick with online grocery the post-covid world of US online grocery growth. https://research.rabobank.com/far/en/sectors/beverages/will-consumers-stick-with-online-grocery.html. Accessed: 2022-07-26.

NewspaperDirect (2024). Newspaperdirect. Accessed: 2024-11-02.

nShift (2022). Delivery time slots and customer satisfaction. Accessed: 2024-11-06.

O'Flaherty, A., Collins, M., & Wiltshire, A. (2022). Queensland's browns plains without electricity after drone food delivery crashes into powerlines. Accessed: 2024-11-07.

Orenstein, I., Raviv, T., & Sadan, E. (2019). Flexible parcel delivery to automated parcel lockers: models, solution methods and analysis. *EURO Journal on Transportation and Logistics*, 8(5), 683–711.

Oyola, J., Arntzen, H., & Woodruff, D. L. (2018). The stochastic vehicle routing problem, a literature review, part I: models. *EURO Journal on Transportation and Logistics*, 7(3), 193–221.

Peura, H. & Yang, S. A. (2023). Capacity expansion in service platforms: Ownership, commitment, and flexibility. *Available at SSRN 3564074*.

Pillac, V., Gendreau, M., Guéret, C., & Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1), 1–11.

Pina-Pardo, J. C., Silva, D. F., & Smith, A. E. (2021). The traveling salesman problem with release dates and drone resupply. *Computers & Operations Research*, 129, 105170.

Pina-Pardo, J. C., Silva, D. F., Smith, A. E., & Gatica, R. A. (2024a). Dynamic vehicle routing problem with drone resupply for same-day delivery. *Transportation Research Part C: Emerging Technologies*, 162, 104611.

Pina-Pardo, J. C., Silva, D. F., Smith, A. E., & Gatica, R. A. (2024b). Fleet resupply by drones for last-mile delivery. *European Journal of Operational Research*, 316(1), 168–182.

Pindyck, R. S. (1982). Adjustment costs, uncertainty, and the behavior of the firm. *The American Economic Review*, 72(3), 415–427.

Pisano, G. P. (2024). How fast should your company really grow? *Harvard business review*, 102(02), 38–45.

Placek, M. (2024). Same-day delivery market size in U.S. 2024-2030. Accessed: 2024-11-02.

Poikonen, S. & Campbell, J. F. (2021). Future directions in drone routing research. *Networks*, 77(1), 116–126.

Powell, W. B. (2009). What you should know about approximate dynamic programming. *Naval Research Logistics (NRL)*, 56(3), 239–249.

Powell, W. B. (2014). Clearing the jungle of stochastic optimization. In *Bridging data and decisions* (pp. 109–137). Informs.

Powell, W. B. (2021). *Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions*. Hoboken, NJ: Wiley.

Powell, W. B. (2022). *Reinforcement learning and Stochastic Optimization*. John Wiley & Sons.

Psaraftis, H. N. (1988). Dynamic vehicle routing problems. *Vehicle routing: Methods and studies*, 16, 223–248.

Psaraftis, H. N., Wen, M., & Kontovas, C. A. (2016). Dynamic vehicle routing problems: Three decades and counting. *Networks*, 67(1), 3–31.

Pureza, V. & Laporte, G. (2008). Waiting and buffering strategies for the dynamic pickup and delivery problem with time windows. *INFOR: Information Systems and Operational Research*, 46(3), 165–175.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Qi, A., Ahn, H.-S., & Sinha, A. (2017). Capacity investment with demand learning. *Operations Research*, 65(1), 145–164.

Ram, A. (2015). UK retailers face high cost of online deliveries. https://www.ft.com/content/516aa75a-a04c-11e5-beba-5e33e2b79e46#axzz4CPxX2fjJ. Accessed: 2022-07-16.

Ranjbari, A., Diehl, C., Dalla Chiara, G., & Goodchild, A. (2023). Do parcel lockers reduce delivery times? evidence from the field. *Transportation Research Part E: Logistics and Transportation Review*, 172, 103070.

Reed, S., Campbell, A. M., & Thomas, B. W. (2022). The value of autonomous vehicles for last-mile deliveries in urban environments. *Management Science*, 68(1), 280–299.

Revadekar, A., Soni, R., & Nimkar, A. V. (2020). QORAl: Q learning based delivery optimization for pharmacies. In *2020 11th International conference on computing, communication and networking technologies* (pp. 1–7).: IEEE.

Reyes, D., Erera, A., Savelsbergh, M., Sahasrabudhe, S., & O'Neil, R. (2018a). The meal delivery routing problem. *Optimization Online*, 6571.

Reyes, D., Erera, A. L., & Savelsbergh, M. W. (2018b). Complexity of routing problems with release dates and deadlines. *European journal of operational research*, 266(1), 29–34.

Ritzinger, U., Puchinger, J., & Hartl, R. F. (2016). A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, 54(1), 215–231.

Roadie (2024). Roadie survey finds 80delivery. Accessed: 2024-11-02.

Rohmer, S. & Gendron, B. (2020). *A guide to parcel lockers in last mile distribution: Highlighting challenges and opportunities from an OR perspective*. Cirrelt Montreal.

Rubin, B. F. (2018). Amazon Key In-Car delivers packages right to your car's trunk. Accessed: 2024-11-19.

Savelsbergh, M. & Sol, M. (1998). Drive: Dynamic routing of independent vehicles. *Operations Research*, 46(4), 474–490.

Savelsbergh, M. & Van Woensel, T. (2016). 50th anniversary invited article—city logistics: Challenges and opportunities. *Transportation Science*, 50(2), 579–590.

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

Schrotenboer, A. H., Broek, M. A., Buijs, P., & Ulmer, M. W. (2021). Fighting the e-commerce giants: Efficient routing and effective consolidation for local delivery platforms. *arXiv preprint arXiv:2108.12608*.

Secomandi, N. (2001). A rollout policy for the vehicle routing problem with stochastic demands. *Operations Research*, 49(5), 796–802.

Secomandi, N. & Margot, F. (2009). Reoptimization approaches for the vehicle-routing problem with stochastic demands. *Operations Research*, 57(1), 214–230.

Sethi, S. & Sorger, G. (1991). A theory of rolling horizon decision making. *Annals of Operations Research*, 29(1), 387–415.

Sethuraman, S., Bansal, A., Mardan, S., Resende, M. G., & Jacobs, T. L. (2024). Amazon locker capacity management. *INFORMS Journal on Applied Analytics*.

Shankland, S. (2022). Amazon drone deliveries coming to texas city as prime air expands. Accessed: 2024-11-06.

Shelbourne, B. C., Battarra, M., & Potts, C. N. (2017). The vehicle routing problem with release and due dates. *INFORMS Journal on Computing*, 29(4), 705–723.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587), 484–489.

Simchi-Levi, D., Chen, X., Bramel, J., et al. (2005). The logic of logistics. *Theory, algorithms, and applications for logistics and supply chain management*.

Simeonova, S. (2024). The insider's guide to milestone-based funding: What every startup needs to know. Accessed: 2025-03-31.

Sitek, P. & Wikarek, J. (2019). Capacitated vehicle routing problem with pick-up and alternative delivery (cvrppad): model and implementation using hybrid approach. *Annals of Operations Research*, 273(1), 257–277.

Soeffker, N., Ulmer, M. W., & Mattfeld, D. C. (2022). Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review. *European Journal of Operational Research*, 298(3), 801–820.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2), 254–265.

Song, Y., Ulmer, M. W., Thomas, B. W., & Wallace, S. W. (2020). Building trust in home services—stochastic team-orienteering with consistency constraints. *Transportation science*, 54(3), 823–838.

Soufani, K. (2001). The role of factoring in financing UK smes: A supply side analysis. *Journal of Small Business and Enterprise Development*, 8(1), 37–46.

Srinivas, S., Ramachandiran, S., & Rajendran, S. (2022). Autonomous robot-driven deliveries: A review of recent developments and future directions. *Transportation research part E: logistics and transportation review*, 165, 102834.

Statista (2021). E-commerce report 2021. https://www.statista.com/study/42335/ecommerce-report/. Accessed: 2022-10-16.

Statista (2025). Main aspects that shoppers would change about the delivery of products purchased online worldwide in 2024. https://www.statista.com/statistics/1274950/global%2Donline%2Dshoppers%2Dwished%2Dchanges%2Don%2Ddelivery/. Accessed: 2025-04-03.

Steinbruch, F. K., Fernandes, B. S., Nascimento, L. d. S., & Zawislak, P. A. (2022). Outsourcing in startups. *Journal of Entrepreneurship in Emerging Economies*, 14(2), 231–251.

Stroh, A. M., Erera, A. L., & Toriello, A. (2022). Tactical design of same-day delivery systems. *Management Science*, 68(5), 3444–3463.

Subramanyam, A., Mufalli, F., Laínez-Aguirre, J. M., Pinto, J. M., & Gounaris, C. E. (2021). Robust multiperiod vehicle routing under customer order uncertainty. *Operations Research*, 69(1), 30–60.

Sutton, R. S. & Barto, A. (2018). *Reinforcement learning: An Introduction*. MIT Press.

Swinney, R., Cachon, G. P., & Netessine, S. (2011). Capacity investment timing by start-ups and established firms in new markets. *Management Science*, 57(4), 763–777.

Tafidis, P., Farah, H., Brijs, T., & Pirdavani, A. (2022). Safety implications of higher levels of automated vehicles: a scoping review. *Transport Reviews*, 42(2), 245–267.

Taylor, T. A. (2002). Supply chain coordination under channel rebates with sales effort effects. *Management science*, 48(8), 992–1007.

Teale, C. (2022). As companies expand drone deliveries, most consumers have trust issues over the idea. Accessed: 2024-11-07.

The Plant Base (2022). Umiami to establish factory at site of former unilever facility. Accessed: 2024-05-23.

Tidhar, R., Hallen, B. L., & Eisenhardt, K. M. (2025). Measure twice, cut once: Unit profitability, scalability, and the exceptional growth of new firms. *Organization Science*, 36(1), 88–120.

Tilk, C., Olkis, K., & Irnich, S. (2021). The last-mile vehicle routing problem with delivery options. *OR Spectrum*, 43(4), 877–904.

Truden, C., Maier, K., Jellen, A., & Hungerländer, P. (2022). Computational approaches for grocery home delivery services. *Algorithms*, 15(4), 125.

Tsay, A. A., Gray, J. V., Noh, I. J., & Mahoney, J. T. (2018). A review of production and operations management research on outsourcing in supply chains: Implications for the theory of the firm. *Production and Operations Management*, 27(7), 1177–1220.

Ulmer, M. (2017). Delivery deadlines in same-day delivery. *Logistics Research*, 10(3), 1–15.

Ulmer, M. W. (2020). Dynamic pricing and routing for same-day delivery. *Transportation Science*, 54(4), 1016–1033.

Ulmer, M. W., Goodson, J. C., Mattfeld, D. C., & Hennig, M. (2019a). Offline–online approximate dynamic programming for dynamic vehicle routing with stochastic requests. *Transportation Science*, 53(1), 185–202.

Ulmer, M. W., Goodson, J. C., Mattfeld, D. C., & Thomas, B. W. (2020). On modeling stochastic dynamic vehicle routing problems. *EURO Journal on Transportation and Logistics*, 9(2), 100008.

Ulmer, M. W. & Savelsbergh, M. (2020). Workforce scheduling in the era of crowdsourced delivery. *Transportation Science*, 54(4), 1113–1133.

Ulmer, M. W. & Streng, S. (2019). Same-day delivery with pickup stations and autonomous vehicles. *Computers & Operations Research*, 108, 1–19.

Ulmer, M. W. & Thomas, B. W. (2018). Same-day delivery with heterogeneous fleets of drones and vehicles. *Networks*, 72(4), 475–505.

Ulmer, M. W., Thomas, B. W., Campbell, A. M., & Woyak, N. (2021). The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. *Transportation Science*, 55(1), 75–100.

Ulmer, M. W., Thomas, B. W., & Mattfeld, D. C. (2019b). Preemptive depot returns for dynamic same-day delivery. *EURO journal on Transportation and Logistics*, 8(4), 327–361.

Van Heeswijk, W. J., Mes, M. R., & Schutten, J. M. (2019). The delivery dispatching problem with time windows for urban consolidation centers. *Transportation Science*, 53(1), 203–221.

Van Mieghem, J. A. (2003). Commissioned paper: Capacity management, investment, and hedging: Review and recent developments. *Manufacturing & Service Operations Management*, 5(4), 269–302.

Vansteenwegen, P., Souffriau, W., & Van Oudheusden, D. (2011). The orienteering problem: A survey. *European Journal of Operational Research*, 209(1), 1–10.

Vegconomist (2022). Umiami raises largest-ever €30m funding amid whole-cut chick'n scale-up. Accessed: 2024-05-23.

Verlinde, S. & Macharis, C. (2016). Who is in favor of off-hour deliveries to brussels supermarkets? Applying Multi Actor Multi Criteria analysis (MAMCA) to measure stakeholder support. *Transportation Research Procedia*, 12, 522–532.

Voccia, S. A., Campbell, A. M., & Thomas, B. W. (2019). The same-day delivery problem for online purchases. *Transportation Science*, 53(1), 167–184.

Wang, W., Ferguson, M. E., Hu, S., & Souza, G. C. (2013). Dynamic capacity investment with two competing technologies. *Manufacturing & Service Operations Management*, 15(4), 616–629.

Wang, Y., Niu, B., Guo, P., & Song, J.-S. (2021). Direct sourcing or agent sourcing? contract negotiation in procurement outsourcing. *Manufacturing & Service Operations Management*, 23(2), 294–310.

Wang, Z., Rahmani, M., & Ramachandran, K. (2024). Advising entrepreneurs: Optimal recommendation of alternatives. *Manufacturing & Service Operations Management*, 26(6), 2015–2029.

Waymo (2024). Waymo self-driving cars are delivering Uber Eats orders for first time. Accessed: 2024-11-02.

Waßmuth, K., Köhler, C., Agatz, N., & Fleischmann, M. (2023). Demand management for attended home delivery — A literature review. *European Journal of Operational Research*, 311(3), 801–815.

Weibel, R. E. (2005). *Safety considerations for operation of different classes of unmanned aerial vehicles in the national airspace system*. PhD thesis, Massachusetts Institute of Technology.

Wikipedia (2025). Britishvolt — wikipedia, the free encyclopedia. Accessed: 2025-03-31.

Wu, H., Herszterg, I., Savelsbergh, M., & Huang, Y. (2023). Service network design for same-day delivery with hub capacity constraints. *Transportation Science*, 57(1), 273–287.

Yang, X. (2020). Double-11 deliveries: As delivery companies compete, couriers lose out. Accessed: 2024-11-05.

Yao, B., McLean, C., & Yang, H. (2019). Robust optimization of dynamic route planning in same-day delivery networks with one-time observation of new demand. *Networks*, 73(4), 434–452.

Yildiz, B. & Savelsbergh, M. (2019a). Provably high-quality solutions for the meal delivery routing problem. *Transportation Science*, 53(5), 1372–1388.

Yildiz, B. & Savelsbergh, M. (2019b). Service and capacity planning in crowd-sourced delivery. *Transportation Research Part C: Emerging Technologies*, 100, 177–199.

Yoo, O. S., Corbett, C. J., & Roels, G. (2016a). Optimal time allocation for process improvement for growth-focused entrepreneurs. *Manufacturing & Service Operations Management*, 18(3), 361–375.

Yoo, O. S., Roels, G., & Corbett, C. J. (2016b). The time–money trade-off for entrepreneurs: When to hire the first employee? *Manufacturing & Service Operations Management*, 18(4), 559–569.

Zhang, J., Luo, K., Florio, A. M., & Van Woensel, T. (2023). Solving large-scale dynamic vehicle routing problems with stochastic requests. *European Journal of Operational Research*, 306(2), 596–614.

Zhang, J. & Van Woensel, T. (2023). Dynamic vehicle routing with random requests: A literature review. *International Journal of Production Economics*, 256, 108751.

Zhang, S., Ohlmann, J. W., & Thomas, B. W. (2018). Dynamic orienteering on a network of queues. *Transportation Science*, 52(3), 691–706.

Zhao, Y., Cattaruzza, D., Kang, N., & Roberti, R. (2024). Synchronized deliveries with a bike and a self-driving robot. *Transportation Science*, 58(1), 219–239.

Zhou, L., Baldacci, R., Vigo, D., & Wang, X. (2018). A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution. *European Journal of Operational Research*, 265(2), 765–778.

Zhu, K., Zhang, R. Q., & Tsung, F. (2007). Pushing quality improvement along supply chains. *Management science*, 53(3), 421–436.

Zipline (2024). Zipline designs, manufactures, and operates the world's largest drone delivery service. Accessed: 2024-11-02.