Designing Services for Meal Delivery Platforms

An Integrated Optimization of Restaurant Selection and Rider Allocation

F.E.M. (Frederiek) Backers

Delft University of Technology

TU Delft

Designing Services for Meal Delivery Platforms

An Integrated Optimization of Restaurant Selection and Rider Allocation

by

F.E.M. (Frederiek) Backers

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Friday March 28, 2025 at 13:45 PM.

Student number:	4704452	
Project duration:	September, 2024 – March, 2025	
Thesis committee:	S. Sharif Azadeh	Chair
(TU Delft)	D. Xia	Supervisor
	M. Y. Maknoon	Supervisor
	P. Stokkink	Additional examiner

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Preface

This thesis marks the completion of my master's and the end of a challenging yet rewarding journey. Throughout this process, I have had the privilege of working with and learning from inspiring individuals, without whom this thesis would not have been possible.

First and foremost, I would like to express my deepest gratitude to my chair, Shadi Sharif Azadeh, for giving me the opportunity to work on this research and for the feedback throughout the process. Though challenging at times, your belief in me has been a driving force in completing this thesis. I also want to sincerely thank Yousef Maknoon and Dongyang Xia for their time, invaluable feedback, and continuous support. Our meetings may never have finished on time, but they were some of the most insightful brainstorming sessions I have ever been a part of. Thank you for pushing me to deliver the best version of my work, I am truly grateful for your encouragement.

Beyond academia, I am incredibly thankful for the support of my friends. Pim, Coen, and Sabine, your company and laughter made these months far more enjoyable. Vera, Sterre, and Sophie, your support meant a lot to me, and I truly appreciate it. To my family and all my other friends, thank you for always being there.

I would also like to extend my appreciation to Patrick Stokkink for being part of my thesis examination committee. Your time and effort are greatly valued.

With this chapter coming to a close, I look forward to what the future holds. If this journey has taught me anything, it is that curiosity and perseverance open doors to unexpected opportunities, and I am excited to see where those lead.

F.E.M. (Frederiek) Backers
Delft, March 2025

Summary

Meal delivery platforms operate in dynamic environments, facing complex challenges including fluctuating customer demand, deciding where and which restaurants to offer, and efficient rider allocation. This thesis introduces the Restaurant Selection and Rider Dimensioning Problem (RSRDP) by proposing an integrated optimization model that jointly determines the optimal restaurant assortments and rider allocations to maximize the platform's expected profitability, while maintaining high service quality standards.

A nested logit model captures customer purchasing behavior on the platform to model how customers choose a restaurant to order from, where customers initially choose a cuisine type and subsequently select a specific restaurant within that type. Rider operations are modeled through a spatial-temporal network, enabling optimized management of rider flows across urban zones. We formulate the problem as a Mixed-Integer Linear Program (MILP) and research advanced solution methods including Benders decomposition, column generation, branch-and-price, and heuristics. Finally, we propose our novel Iterative Assortment Generation (IAG) algorithm to solve the problem, ensuring computational feasibility for large-scale scenarios.

Computational experiments based on simulated urban delivery data illustrate that integrating restaurant selection with rider dimensioning consistently increases profitability and maintains high service quality compared to traditional methods that separate these decisions. Furthermore, we explore the impact of two distinct rider compensation policies: commission-based (payment per delivery) and fixed employment (hourly wages). Our findings reveal significant trade-offs between these compensation structures. Commission-based compensation offers greater operational flexibility, adaptability to fluctuating demands, and cost-effectiveness; however, it also leads to increased rider relocations and potentially reduced rider satisfaction due to uncertainty in workload. Conversely, fixed employment provides stable rider availability, promoting better workforce management, but potentially incurs higher operational costs and decreased responsiveness to short-term demand fluctuations.

Based on these insights, we recommend meal delivery platforms adopt integrated optimization models as part of their tactical planning to effectively balance service quality and profitability. Platforms should carefully evaluate their rider compensation policies, recognizing that the ideal choice may depend on the specific context of their operational environment, such as demand variability, competitive landscape, and workforce preferences.

Future research could build on this study by investigating dynamic compensation schemes that adapt in real-time to observed demand conditions. Additionally, incorporating hybrid compensation schemes, combining salaried and commission-based riders, and utilizing the strengths of both policies, provides a promising research direction. Developing multi-objective models that balance profitability against service quality measures, such as timely deliveries or rider satisfaction, would further guide operational decisions and customer and rider retention. Furthermore, future research could improve the way service districts are designed. Instead of using fixed boundaries, districts could be adjusted dynamically and tailored more precisely to specific customer segments and changing demand patterns. This approach would mean that restaurant offerings can also change throughout the day to closely match customers' preferences, potentially increasing overall profitability. Furthermore, modeling couriers individually, rather than as aggregated flows, allows more precise scheduling that better accounts for realistic travel times, shift durations, and rider constraints. Although this approach increases modeling complexity and computational demands, it offers more accurate operational insights, leading to better-informed management decisions. Finally, employing heuristic-driven column-generation methods and comparing solution approaches systematically can improve scalability and accuracy.

Contents

1 Introduction

Meal delivery platforms operate in an uncertain environment, where customers arrive to the system and place their order, which must be fulfilled in a timely manner. These platforms aim to achieve some key quality service level elements: ensuring on-time delivery to maintain customer satisfaction, providing a diverse selection of restaurants to attract demand, and managing operational costs to remain profitable. These elements translate to the challenges these platforms face: the efficient dimensioning of riders, who must be engaged based on their spatial and temporal distribution to meet fluctuating demand, and which restaurants to offer on the platforms so as to maximize expected profit while offering diverse options.

Existing research on meal delivery platforms primarily focuses on either operational efficiency or demand-side management, but rarely integrates both aspects. Prior studies can be grouped into two main categories. The first category focuses on the operational side, optimizing routing costs and reducing delivery times [Xue et al., 2021, Kancharla et al., 2024, Ulmer et al., 2021], offering valuable insights into fleet operations and dispatching strategies. However, these studies generally assume that customer demand is exogenous and fixed, meaning the platform has no control over how demand can be shaped. Other works examine related operational challenges such as delivery time estimation, rider shift scheduling, and courier imbalances [Liu et al., 2018, Tang et al., 2016]. The second category focuses on restaurant selection, which is mainly studied in the domain of assortment optimization, aiming for revenue maximization of offering products to customers. These studies, however, neglect the operational implications of assortment decisions on rider dimensioning.

The central research question guiding this thesis is defined as: *How can we design the services for a meal delivery platform to maximize expected profits and solve it for large-scale systems?* A key insight of this research is that meal delivery platforms function as interconnected systems where each operational decision influences platform performance. Customers expect timely deliveries, while riders require manageable workloads, and platforms must remain profitable. Unlike previous studies that address these elements in isolation, we recognize that the interactions among them create significant research opportunities. While customer arrival is exogenous, platform controlled decision-making, such as restaurant selection and rider dimensioning, and their interaction, are endogenous and can shaped by designing the services. In this research, we define and present a model for the *Restaurant Selection and Rider Dimensioning Problem* (RSRDP), which jointly determines restaurant selection and rider dimensioning to maximize expected platform profitability while ensuring high service levels. Profit is defined as restaurant commission revenue minus rider costs. We compare two compensation policies for the rider costs: commission-based (CB), where riders get paid per delivery completed, and fixed employment (FE), where riders receive hourly wages. We evaluate these policies to assess their impact on costs, rider availability, and service quality.

To solve the RSRDP, we propose a novel iterative assortment generation heuristic, inspired by column generation, to iteratively generate and evaluate candidate assortments of restaurants and efficiently converge to a good solution. Additionally, we apply Benders decomposition to decouple assortment decisions and rider dimensioning, accelerating convergence in large instances. Our results demonstrate that integrating assortment and rider dimensioning decisions consistently enhances platform profitability while maintaining service quality, including timely deliveries and balanced rider workloads. Furthermore, we evaluate the impact of the two rider compensation policies, finding that the commission-based approach improves operational flexibility but increases rider relocations, whereas the fixed employment scheme ensures a stable required workforce.

1.1. Scientific and societal relevance

From an academic perspective, the contributions of this research to the field of operations research and on-demand service design are twofold. First, it extends the body of knowledge on integrated optimization by formulating a model that simultaneously addresses assortment and rider dimensioning decisions. Second, it explores the scalability of the model to accommodate large-scale systems by proposing a novel efficient computational method, tailored to the problem's inherent complexity. The approach not only enhances the theoretical understanding of on-demand service systems but also provides novel methodological contributions that can be applied to other fields of urban logistics, such as last-mile delivery for retail and grocery sectors, or emergency response and healthcare services.

From a societal perspective, the outcomes of this research have the potential to significantly improve the efficiency and reliability of meal delivery platforms, which have become an essential service in modern urban environments. Enhancing operational efficiency and ensuring timely deliveries directly contribute to improved customer satisfaction. Moreover, by optimizing rider scheduling and developing effective compensation strategies, this research supports more sustainable and equitable labor practices. Such improvements in service reliability and workforce management can lead to reduced environmental impacts through optimized routing and diminished operational waste, ultimately fostering more resilient urban communities.

Furthermore, the broader implications of this research extend into policy-making and tactical business planning within the rapidly evolving landscape of on-demand services. The integrated framework developed here offers actionable managerial insights for platform designers, informing decisions that balance profitability with high service quality. These findings provide a robust foundation for future innovations in smart urban logistics, paving the way for more adaptive, sustainable, and customer-centric service models. In doing so, the study not only contributes to academic discourse but also serves as a resource for driving positive societal change in the context of urban mobility and service delivery.

1.2. Methodology

The proposed methodology is founded on the development of an integrated optimization model that simultaneously addresses restaurant assortment selection and rider dimensioning in the context of meal delivery platforms. The methodology employs a nested logit model to capture the intricacies of customer decision-making processes on the platform, ensuring that restaurant offerings are optimally aligned with consumer preferences. The decision on which restaurants are opened, given the location of the customer, is integrated with a framework that decides when and where to deploy riders. The optimization framework is formulated as a Mixed-Integer Programming model (MIP). To solve the problem, we investigate exact and approximation approaches, including branch-and-price, Benders decomposition and heuristics.

To ensure the robustness and practical relevance of the model, numerical experiments will be conducted using simulated urban demand scenarios, based on available open-source industry information. The scope of the project encompasses both tactical, restaurant assortment optimization, and operational, rider dimensioning, levels of meal-delivery service design. The geographical focus is on urban areas where (bicycle) riders are prevalent. The temporal scope emphasizes short-term daily operations, suitable for capturing rapid fluctuations in order volume and peak demand times.

The study pursues the following objectives:

1. Develop a Mixed-Integer Programming (MIP) model for the Restaurant Selection and Rider Dimensioning Problem (RSRDP).
2. Design and implement efficient solution methodologies to solve the RSRDP for both small-scale and large-scale problem instances, ensuring computational tractability.
3. Analyze the managerial implications of the proposed model, particularly in assessing the impact of different compensation policies on network performance, service quality, and platform profitability.

1.3. Document structure

The remainder of this thesis is structured as follows. Chapter 2 reviews related literature. Chapter 3 presents the conceptual representation, laying the groundwork for the mathematical formulation presented in Chapter 4. Chapter 5 outlines the resolution approaches, with computational results and managerial insights outlined in Chapter 6. Finally, Chapter 7 concludes the study with recommendations for future research.

2

Related literature

Meal delivery platforms have reshaped urban food consumption with on-demand, app-based services. They combine transportation logistics with revenue management, requiring decisions that balance customer satisfaction with profitability. Customer satisfaction depends on timely deliveries, high service quality, and ample restaurant options, while profitability stems from cost-efficiency and strategic pricing. To better understand the complexities of meal delivery platforms, we follow the meal delivery process as depicted in Figure 2.1, examining the perspectives of the customers, orders, and riders at each stage, along with the associated challenges and how current literature tackles these.

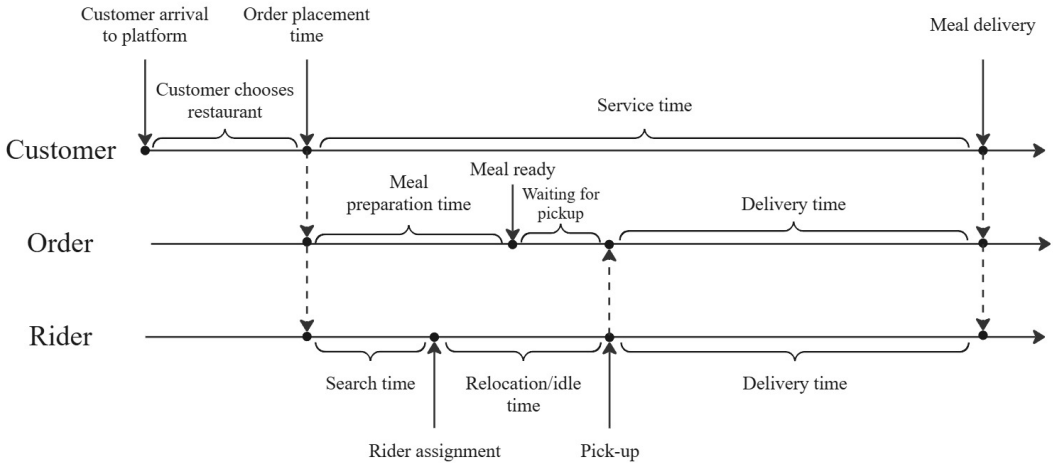

Figure 2.1: Ordering process timeline from customer, order and rider perspective.

2.1. Request arrival and customer decisions

A primary challenge for meal delivery platforms is managing the dynamic arrival of orders. Customer arrivals are inherently unpredictable, and platforms often model customer arrivals as stochastic processes. For instance, Xue et al. [2021] use an empirical distribution based on real observed data, while Kancharla et al. [2024] adopt a Poisson process. The meal delivery industry shares this challenge with on-demand micro-mobility services. However, unlike services where requests can be declined, meal delivery platforms must fulfill all incoming orders. The study by Li and Wang [2024] incorporates probabilistic demand modeling in mobility-on-demand services, demonstrating its relevance in dynamic dispatching. These approaches align with our work, where we model arrivals of requests probabilistically using a Poisson process.

Once customers are on the platform, they navigate a two-stage decision process. First, based on personal tastes, dietary needs, and even mood, customers select a preferred cuisine category. Next, restaurants offering that cuisine are presented, where the choice is influenced by factors such as estimated delivery time, pricing, and restaurant reputation [Fakfare, 2021, Chua et al., 2020]. This underscores the importance of curating restaurant offerings that align with consumer tastes, including more heterogeneity customized to the individual, also suggested by Aparicio et al. [2025]. Our work builds on these insights by integrating customer choice modeling

using a nested logit model as presented by Davis et al. [2014] to aid decision-making on which restaurants to offer.

Once a customer places an order, meal preparation begins at the restaurant. Meal preparation times are uncertain, as they depend on kitchen workload, meal complexity, and restaurant efficiency. This uncertainty can create delays, affecting rider scheduling and customer satisfaction. Ulmer et al. [2021] examine uncertain meal preparation times and propose buffering techniques to minimize delays. Their anticipatory customer assignment approach optimizes real-time order bundling and dispatching. While our model does not explicitly incorporate meal preparation uncertainties, it introduces flexibility in rider assignment to accommodate variability in meal readiness times. This flexibility acts as a buffer for uncertainties in general, ensuring that unexpected delays in preparation do not significantly disrupt delivery operations.

2.2. Rider assignment and supply-demand management

Once an order is placed, matching it with an available rider poses another challenge. The spatial distribution of orders and riders requires quick, efficient dispatch. To address this, Liu et al. [2018] incorporate predictive travel time analytics into their order assignment model, while Li et al. [2024] improve spatial efficiency using a dynamic matching radius. Additionally, the compensation structure of the riders affects rider availability. Ke et al. [2022] explore how different wage schemes impact service quality and profitability, demonstrating that optimized pay schemes influence rider participation and order fulfillment success rates. Rider participation is also studied by Tang et al. [2016], who demonstrate that while higher wages can attract more riders, they might also lead to inefficiencies during low-demand periods. Our research evaluates two static compensation policies, commission-based and fixed employment, to understand their affect on both service quality and profitability.

2.3. Delivery efficiency

After rider assignment, ensuring prompt delivery is key to ensuring service quality. This challenge is addressed in various ways: Liu and Luo [2023] propose a stochastic dynamic driver dispatching system, optimizing routing through Benders decomposition, while Yildiz and Savelsbergh [2019a] explore multi-objective optimization to balance cost and service quality. Their findings also highlight that compensation schemes and courier schedules play a crucial role in ensuring service reliability. Additionally, Carlsson et al. [2021] investigate how geographic familiarity impacts rider efficiency, proposing a partitioning algorithm to optimize delivery regions. Another upcoming method to increase efficiency includes order bundling within a single trip, Steever et al. [2019] examine bundling strategies that aim to minimize delays while maintaining routing efficiency. However, Yildiz and Savelsbergh [2019a] analyze the trade-offs between bundling efficiency and service quality, concluding that bundling does not always yield cost savings yet can decrease service quality. Our model does not include bundling but instead focuses on maintaining strict delivery windows to ensure high service quality and timely fulfillment, as assumed in other studies [Ulmer et al., 2021, Kancharla et al., 2024, Li et al., 2022].

Routing efficiency can also be enhanced through strategic rider relocation. Bell et al. [2024] propose a Markov chain-based relocation model where couriers circulate through the city in a structured manner, optimizing transitions based on demand probabilities. Yang et al. [2024] examine how freelance drivers make routing and dispatch decisions based on probabilistic acceptance behavior. While their work focuses on mobility services, it provides insights into how supply-side constraints impact service quality and profitability. In our model, we improve delivery efficiency by strategically relocating couriers based on future demand predictions, ensuring that riders are positioned optimally before orders arrive. Because we can shape demand distribution through assortment optimization of restaurant offerings, we can exert greater control over the spatial distribution of arriving orders.

2.4. Rider scheduling

Beyond real-time rider dispatching, platforms must manage workforce scheduling. Platforms can hire riders as employees or engage them as freelancers. Employed riders provide stability and better workforce planning, but they increase fixed operational costs. Freelance riders offer flexibility, but their availability is uncertain and influenced significantly by compensation structures. Ulmer and Savelsbergh [2020] explore a hybrid workforce model that incorporates both scheduled and unscheduled riders. Their work highlights the importance of structured scheduling while allowing flexibility through crowdsourced labor. The compensation policies implied by the workforce models play a crucial role in rider scheduling. Yildiz and Savelsbergh [2019b] examine how service

radius adjustments impact profitability by balancing rider costs and restaurant commissions. They also incorporate restaurant availability as a function of the service radius, allowing the platform to influence demand through spatial adjustments. However, their study assumes exogenous demand and self-scheduling riders who can reject orders, whereas our approach assumes riders must adhere to platform decision-making on rider movements and incorporates restaurant availability through assortment optimization.

2.5. Our contributions

The analysis of current literature reveals that existing literature predominantly addresses restaurant assortment decisions separately from rider dimensioning, overlooking the fact that decisions on restaurant offerings determine the spatial demand distribution of orders, which in turn influences required rider deployment. Our work advances the literature by proposing an optimization approach that simultaneously determines the optimal rider dimensioning and the assortment of restaurants being offered to the customer. Additionally, we evaluate static compensation policies and their impact on system efficiency, offering insights into how wage policies influence workforce management and platform profitability. Finally, we propose a novel methodology to solve the problem, ensuring practical applicability for larger problem instances. In Chapter 3, we explain in detail the conceptual representation of this problem.

3

Conceptual representation

This chapter introduces the notation and core concepts needed for the RSRDP formulation presented in Chapter 4. We structure the problem in a spatial-temporal network and explain the interplay between revenue, costs, customer demand, assortment selection, and rider dimensioning.

3.1. Service design structure

To formalize the platform operations, we consider a pre-defined operating area and a nominal day for planning. Our approach employs a two-level spatial representation by dividing this area into larger service districts and smaller hexagonal zones. Service districts, denoted by $d \in D$, capture market heterogeneity by grouping regions with distinct customer preferences and socio-economic traits. These districts also define the curated set of restaurants visible to customers, enabling strategic demand shaping through assortment decisions that directly influence order distribution. To manage rider allocation at a more granular level, the area is further partitioned into hexagonal zones, denoted by $m \in M$. These uniformly distributed hexagons capture rider movement dynamics and allow for accurate travel time computation. Riders travel between zones to complete deliveries and can be strategically relocated to balance supply and demand, thereby minimizing delivery times and operational costs. The interaction between service districts and zones introduces spatial-temporal dependencies that our framework explicitly models, capturing the feedback loop between assortment-driven demand generation and rider management.

We let κ be the travel time between adjacent zone centroids. The travel time between any two zones m and m' is given by $\tau_{mm'}$, which depends on κ and the shortest path distance. It is assumed that intrazonal travel time τ_{mm} equals κ, relating to the maximum travel distance within a zone from the hexagonal structure. Additionally, to discretize the day into manageable intervals, we define $t \in T = \{0, \kappa, ..., T_{max}\}$ where κ also serves as the length of each time period. This synchronized spatial-temporal discretization enables us to track the progression of demand and rider movements over the course of the time horizon. Figure 3.1 presents an example of the hexagonal zone structure on the left, and the service districts by the various colors on the right.

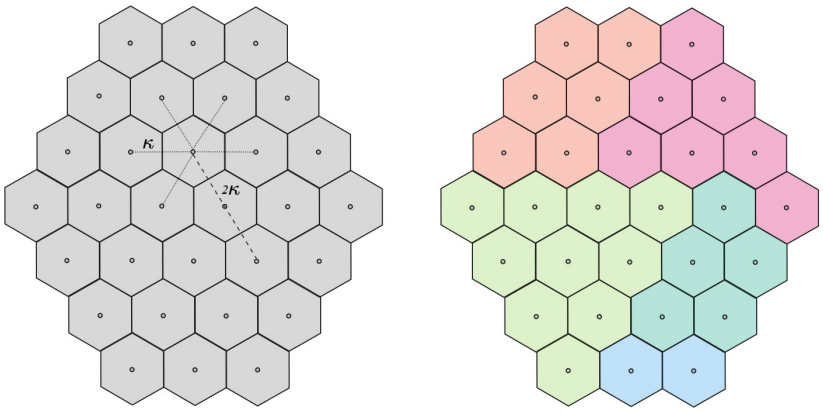

Figure 3.1: Hexagonal zone structure with travel time κ between adjacent zones (left) and service districts $d \in D$ (right), presented by different colors including one or multiple zones $m \in M$.

3.2. Courier network activity

To present courier movement over the planning horizon $|T|$, we construct the spatial-temporal network $G(N, A)$ where each node $(m, t) \in N$ represents a zone $m \in M$ at period $t \in T$. Using arc-based notation with directional arcs, we define the movement of couriers between zones over time, transitioning from an earlier period to a later one. Riders are assumed homogeneous, and all travel at the same constant speed. The network arcs represent rider flow and are defined as arcs $a \in A$ with $a = ((m, t), (m', t + \tau_{mm'}))$ such that $\tau_{mm'}$ is the travel time between zone m and m'. The flow on these arcs represent couriers traveling from zone m starting at time t to a zone m' arriving at time $t + \tau_{mm'}$, defined for all combinations of zones m and m' and periods $t \in [0, T_{max} - \tau_{mm'}]$. Figure 3.2 illustrates these arcs for a three-zone-four-period network.

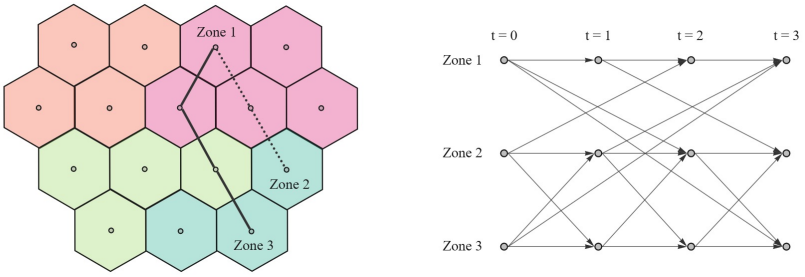

Figure 3.2: Hexagonal zone structure with colored service districts (left) and three-zone-four-period spacial-temporal network example where the arcs represent possible courier flows.

Because all riders are assumed to travel at a constant speed, each arc reflects a possible flow from an earlier time to a later one, and for every zone and period, riders can travel from and to adjacent nodes within the spatial-temporal network. The sets of outgoing and incoming arcs at node (m, t) are denoted by $A^+_{(m,t)}$ and $A^-_{(m,t)}$, respectively, and we do not impose capacity constraints on the arcs. Figure 3.3 presents an example for the origin and destination adjacent arc sets.

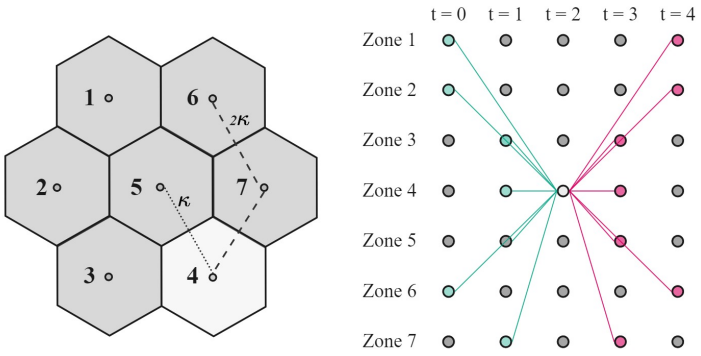

Figure 3.3: Example of adjacent nodes for zone 4 structure. Origin arcs $A^-_{(4,2)}$ are presented in blue and destination arcs $A^+_{(4,2)}$ are presented in magenta.

3.3. Customer arrivals and delivery timeline

Customer arrivals are treated as requests and are aggregated over zones. Specifically, arrivals in zone m at period t denoted λ_{mt} follow a Poisson distribution, capturing the natural fluctuations in demand. We model $\lambda_{mt} \sim Poisson(rate)$, where $rate = Base(t) \cdot (1 + \epsilon_{mt})$. Here, $Base(t)$ is a shape function that rises and falls with typical meal times, and ϵ_{mt} introduces normally distributed random fluctuations. A request becomes an order if the customer decides to purchase a meal from an available restaurant showcased in the assortment for the district in which the customer is located. How the customer chooses the restaurant is explained in the subsequent section.

Each order placed at time t on the platform must be fulfilled within a delivery window $[t, t + \rho]$ Practically, this window includes meal preparation time and rider travel time, plus a small buffer to accommodate unexpected

delays. If η represents the average meal preparation time, and $\tau_{mm'}$ is the travel time from a restaurant located in zone m to the customer's zone m', then the rider must pick up the meal after η and complete the delivery before the total elapsed time reaches ρ. Figure 3.4 illustrates the possible arcs that may be utilized to satisfy the demand, allowing for some flexibility and possible higher efficiency by providing multiple options in some scenarios.

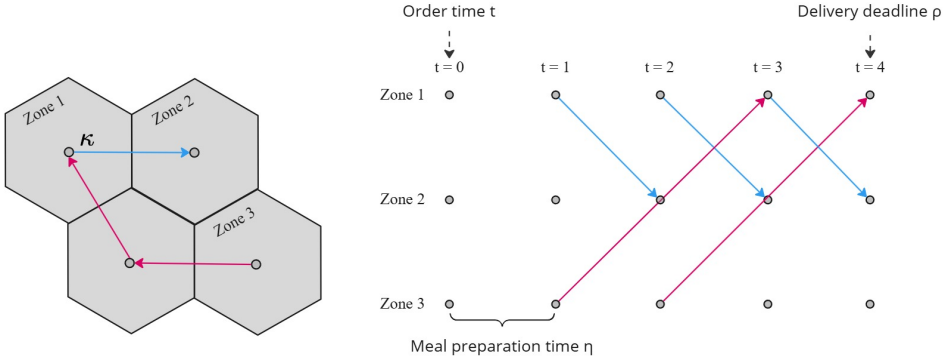

Figure 3.4: Illustration of potential courier flow along arcs to meet demand. Orders from Zone 1 to Zone 2 are fulfilled via the blue arcs, while orders from Zone 3 to Zone 1 follow the magenta arcs. Meal preparation time η and the maximum delivery deadline ρ are accounted for.

3.4. Service district assortment optimization

In addition to managing rider activity, the platform chooses which restaurants to offer in each service district, from which customers can order. Let R be the set of all restaurants in the system, and let $r \in R$. A district d may only include restaurants that can reliably deliver within the delivery window ρ. Specifically, for a customer located in zone m and a restaurant in zone m', $R_m = \{r \in R : \tau_{m'm} \leq \rho - \eta\}$ denotes the set of restaurants capable of serving zone m, and if $b_m^d \in \{0, 1\}$ indicates whether zone m is covered by district d, then $R_d = \bigcap_{m \in M : b_m^d = 1} R_m$ is the set of restaurants that can fulfill the delivery window requirements for all zones in district d. Finally, we define a finite set of cuisine types Q, such that $q \in Q$ and each restaurant $r \in R$ belongs to exactly one cuisine type, administered by the parameter $e_r^q \in \{0, 1\}$ that is one when restaurant r is of cuisine q. In this way, the set $R_q^d = \bigcap_{m \in M : b_m^d = 1} \{r \in R_m : e_r^q = 1\}$ represents the set of restaurants that may be included for the assortment of district d for cuisine type q. The model may decide which of these restaurants to offer. We denote the final assortment of restaurants for cuisine type q in service district d by $S_q^d \subseteq R_q^d$.

The model may include constraints on the assortment to align with business or operational goals. For instance, a minimum number of restaurants of each cuisine type may need to be included in a service district to meet marketing or customer satisfaction requirements. Let K_q^d denote the minimum number of restaurants of cuisine type q required in the assortment for service district d. The binary parameter e_r^q equals one if restaurant r belongs to cuisine type q and zero otherwise. Each restaurant is assigned to a single cuisine type to maintain the nested structure and adhere to probability theory. By incorporating such constraints, the platform ensures compliance with these operational requirements while optimizing the assortment.

3.5. Customer purchasing behavior

To model how customers choose restaurants from the platform's assortment, we employ a nested choice framework inspired by Davis et al. [2014]. On meal delivery platforms, customers tend to first choose a cuisine type and then select a restaurant within that category. The nested logit model follows the same structure: customers first choose a nest (cuisine) and thereafter a product within that nest (restaurant). We define v_{qr}^d as the attraction value of restaurant r of cuisine q within district d. This value encapsulates all relevant utility parameters (e.g. price, quality, proximity) and is assumed known from prior choice modeling research or historical data. We also define v_{q0}^d as the attraction of the no-purchase options within cuisine q for district d. Suppose the assortment of cuisine q in district d is $S_q^d \subseteq R_q^d$. The total attraction in nest q is then $V_q^d(S_q^d) = v_{q0}^d + \sum_{r \in S_q^d} v_{qr}^d$. Under the framework, the probability that a customer orders from restaurant $r \in S_q^d$ given they have selected nest q

is:

$$\mathbb{P}^d_{r|q}(S^d_q) = \frac{v^d_{qr}}{v^d_{q0} + \sum_{r \in S^d_q} v^d_{qr}} = \frac{v^d_{qr}}{V^d_q(S^d_q)} \tag{3.1}$$

Let p^d_{qr} represent the revenue from an order at restaurant r in cuisine nest q for district d. The expected revenue from customers ordering within nest q is:

$$\pi^d_q(S^d_q) = \frac{\sum_{r \in S^d_q} p^d_{qr} v^d_{qr}}{V^d_q(S^d_q)} = \sum_{r \in S^d_q} p^d_{qr} \cdot \mathbb{P}^d_{r|q}(S^d_q) \tag{3.2}$$

Each cuisine nest q has a dissimilarity parameter $\gamma^d_q \geq 0$, which accounts for the degree of dissimilarity of the restaurants within the nest. We assume that these parameters are also researched a priori. Let v^d_0 denote the attraction value of the no-purchase option for choosing any of the nests in district d. If we offer assignments $(S^d_1, ..., S^d_{|Q|})$ over all nests with $S^d_q \subseteq R^d_q \; \forall q \in Q, \; d \in D$, then a customer chooses nest q in district d with probability:

$$\mathbb{P}^d_q(S^d_q) = \frac{V^d_q(S^d_q)^{\gamma^d_q}}{v^d_0 + \sum_{q \in Q} V^d_q(S^d_q)^{\gamma^d_q}} \tag{3.3}$$

Then the probability of choosing restaurant r of cuisine q in district d is given by:

$$\mathbb{P}^d_{qr}(S^d_q) = \mathbb{P}^d_q(S^d_q) \cdot \mathbb{P}^d_{r|q}(S^d_q) = \left(\frac{V^d_q(S^d_q)^{\gamma^d_q}}{v^d_0 + \sum_{q \in Q} V^d_q(S^d_q)^{\gamma^d_q}} \right) \cdot \left(\frac{v^d_{qr}}{V^d_q(S^d_q)} \right) = \frac{v^d_{qr} \cdot V^d_q(S^d_q)^{\gamma^d_q - 1}}{v^d_0 + \sum_{q \in Q} V^d_q(S^d_q)^{\gamma^d_q}} \tag{3.4}$$

And the expected revenue from customers ordering within district d is:

$$\Pi^d(S^d_1, ..., S^d_{|Q|}) = \sum_{q \in Q} \mathbb{P}^d_q(S^d_q) \pi^d_q(S^d_q) = \frac{\sum_{q \in Q} \pi^d_q(S^d_q) V^d_q(S^d_q)^{\gamma^d_q}}{v^d_0 + \sum_{q \in Q} V^d_q(S^d_q)^{\gamma^d_q}} \tag{3.5}$$

By modeling customer choices in this manner, we can estimate the likelihood of each restaurant being selected by customers located in each service district, which subsequently informs the effective flow of orders in the network, which we relate to the number of needed riders to deliver these orders within the network. Equation (3.5) highlights the independence of assortments across service districts. The only interaction between service districts in this problem arises from riders needed to deliver orders between different zones belonging to different districts. The assortment offered in each district dictates the available restaurants, which in turn influences the customer demand generated at those restaurants.

3.6. Rider dimensioning

Using the customer behavioral probabilities derived, we can estimate the expected number of orders between zones. For each period t and zone m, we determine how many orders should be delivered from zone m' to m and therefore how many couriers are required. Suppose that the set $\hat{R}_m$ denotes the restaurants located in zone m. Given the number of requests λ_{mt} and the probability of choosing restaurant r of cuisine q in district d given by Equation (3.4), we denote the number of orders from restaurants located in zone m to customers located in zone m' starting at time t by $\delta^t_{mm'}$. Then:

$$\delta^t_{mm'} = \sum_{d \in D} \sum_{q \in Q} \sum_{r \in \hat{R}_m \cap R^d_q} \lambda_{m't} \cdot \mathbb{P}^d_{qr}(S^d_q) \tag{3.6}$$

As the calculation incorporates probabilities, the number of orders may result in fractional values, representing an average demand across zones. While we assume that each courier can deliver only one order at a time, allowing fractional courier flows in this tactical decision-making model is not only practical but also analytically beneficial. Let the decision variables $u^{in}_{(m,t)}$ and $u^{out}_{(m,t)}$ represent the number of couriers entering and leaving the system node (m, t) in the spatial-temporal network respectively, essentially functioning as source and sink nodes for the couriers. To determine the total fractional number of couriers needed, we sum $u^{in}_{(m,t)}$ across all nodes: $\sum_{(m,t) \in N} u^{in}_{(m,t)}$. Couriers are permitted to enter the system at node, providing the flexibility needed to

optimally meet varying demand patterns. However, we impose constraints to ensure that, on average, couriers work for at least a minimum shift duration θ_{min} and do not exceed a maximum shift duration θ_{max}.

While it would be ideal to track the shift duration of individual couriers, incorporating an "entry time" index into the model would significantly increase its complexity and computational requirements, therefore, we opt to monitor the average shift duration of couriers instead of tracking each individually. However, if one would prefer the model is flexible to include this additional dimension by incorporating an index t_{entry} for the flows and changing the constraints to include this option. This approach allows us to enforce labor regulations and operational policies without the need for granular tracking.

3.7. Profit structure

The objective is to maximize the platform's total expected profit, defined as the difference between revenue and costs. Revenue is obtained through commission paid by restaurants on each order made by customers, based on a percentage of the order value. Each order placed by a customer therefore corresponds to a revenue which can vary depending on the restaurant r and cuisine q and district d. We denote this revenue by p^d_{qr}.

Costs are related to riders and consist of two parts: expenses related to the compensation of riders, and the expenses related to overhead costs depending on the number of couriers that are used in the system. We consider two compensation policies to compare and test the influence of different policies on the performance of the system. The first policy is defined as Commission Based (CB), where riders are compensated for each order they deliver. We define this cost as c^d_{qr}, similar to the profit structure. Similarly, the expected compensation costs of delivering an order within district d related to the CB policy can be found using Equations (3.2) and (3.5) and replacing p^d_{qr} with c^d_{qr}:

$$\hat{c}^{comp}_{CB} = \frac{\sum_{q\in Q}\sum_{r\in S^d_q} c^d_{qr}\cdot \mathbb{P}^d_{r|q}(S^d_q)V^d_q(S^d_q)^{\gamma^d_q}}{v^d_0 + \sum_{q\in Q} V^d_q(S^d_q)^{\gamma^d_q}} \tag{3.7}$$

The second policy is defined as Fixed Employment (FE), where riders are hired by the platform and get compensated a fixed wage per hour. Let c^t be the discretized wage per time unit for a rider. Then traveling along an arc $a = ((m,t),(m',t+\tau_{mm'})) \in A$ incurs a cost $c_a = c^t \cdot \tau_{mm'}$. Summing this over all arcs provides the total wage payout. We generalize the compensation costs for the policies to $c^{compensation}_{policy}$.

The overhead costs exist for both policies, however the overhead costs related to the FE policy are bigger than for the CB policy, as naturally there are higher costs incurred when hiring riders. We define these costs as $c^{overhead}_{policy}$ per rider such that $policy \in \{CB, FE\}$ and calculate the costs based on the number of incoming couriers within the spatial-temporal network:

$$\hat{c}^{overhead}_{policy} = c^{overhead}_{policy} \cdot \sum_{(m,t)\in N} u^{in}_{(m,t)} \tag{3.8}$$

The conceptual framework presented in this chapter underpins the decision-making mechanisms of the RSRDP. We explored the spatial-temporal structure of the network, rider dynamics, customer ordering behavior, and the optimization of restaurant assortments within service districts. These components collectively form the foundation for understanding the interactions between customer demand, restaurant offerings, and rider operations. This framework serves as the stepping stone for the next chapter, where we formalize these concepts into a mathematical formulation.

4

Mathematical formulation

This chapter presents the mathematical formulation of the RSRDP. We first present the formulation in its full non-linear form, whereafter we propose a mathematical formulation that enumerates all possible assortments for each service district d and cuisine nest q, which we can further linearize to obtain a Mixed-Integer Linear Program (MILP). All relevant sets, parameters, and variables can be found in Tables 4.1 and 4.2.

Table 4.1: Sets and parameters used in the mathematical formulations.

Sets and indices		
D	set of service districts	$d \in D$
R	set of restaurants	$r \in R$
Q	set of cuisine types	$q \in Q$
R_m	set of restaurants within delivery limit of zone m	$r \in R_m$
$\hat{R}_m$	set of restaurants located in zone m	$r \in \hat{R}_m$
R_d	set of restaurants within delivery limit for service district d	$r \in R_d$
R_q^d	set of restaurants within delivery limit for cuisine q and service district d	$r \in R_q^d$
M	set of zones	$m, m' \in M$
T	set of time periods such that $T = \{0, \kappa, 2\kappa, ..., T_{max}\}$	$t, t' \in T$
N	set of spatial-temporial nodes (m, t) such that $m \in M, t \in T$	$(m, t) \in N$
A	set of arcs	$a \in A$
$A^+_{(m,t)}$	set of possible destination arcs from node (m, t)	$(m', t') \in A^+_{(m,t)}$
$A^-_{(m,t)}$	set of possible origin arcs to node (m, t)	$(m', t') \in A^-_{(m,t)}$
Parameters		
$\tau_{mm'}$	travel time from zone m to m'; interzonal travel time $\tau_{mm} = \kappa$	[periods]
ρ	delivery time deadline	[periods]
η	meal preparation time	[periods]
λ_{mt}	number of customer arrivals in zone m and period t	[orders]
$c^{overhead}_{policy}$	overhead cost per required rider for business policies	[euro]
c_a	courier salary on arc a	[euro]
c^t	discretized salary per period t	[euro]
c^d_{qr}	cost per service of delivery ordered from restaurant r of cuisine q in district d	[euro]
p^d_{qr}	expected revenue per order at restaurant r for cuisine type q for service district d	[euro]
v^d_{qr}	attraction value of restaurant r for cuisine type q for service district d	[attraction]
v^d_0	attraction value no-purchase option cuisine level for service district d	[attraction]
v^d_{q0}	attraction value no-purchase option restaurant level for cuisine q and district d	[attraction]
γ^d_q	dissimilarity parameter for cuisine type q for service district d	[dissimilarity]
θ_{min}	average minimum shift duration	[periods]
θ_{max}	average maximum shift duration	[periods]
K^d_q	min. number of restaurants of cuisine q to be included in service district d	[#]
e^r_q	binary parameter indicating if restaurant r is of cuisine q	[binary]
b^d_m	binary parameter indicating if zone m is covered by service district d	[binary]

Table 4.2: Variables used in the mathematical formulations.

Variables		
z^d_{qr}	if restaurant r included in assortment for cuisine nest q in service district d	[binary]
$z_{S^d_q}$	if assortment S^d_q offered for cuisine nest q for service district d	[binary]
w_a	courier flow on arc a	[continuous]
$u^{in}_{(m,t)}$	number of couriers starting their work at period t in zone m	[continuous]
$u^{out}_{(m,t)}$	number of couriers leaving the system at period t in zone m	[continuous]
$y_d;\ f_d;\ g_d$	auxiliary variables for model formulation	[continuous]
$x_{md};\ h_{md}$	auxiliary variables for model formulation	[continuous]
$l_{S^d_q};\ k^m_{S^d_q}$	auxiliary variables for model formulation	[continuous]

4.1. Full non-linear formulation

We introduce binary decision variable z^d_{qr} which equals one if restaurant r is included in the assortment for cuisine nest q in service district d and zero otherwise. Looking back at Section 3.5, the customer purchasing behavior is based on the chosen assortment $S^d_q \subseteq R^d_q$. We can represent this assortment using decision variable z^d_{qr} as $S^d_q = \{r \in R^d_q : z^d_{qr} = 1\}$. This changes the total attraction of nest q to $V^d_q(S^d_q) = v^d_{q0} + \sum_{r \in R^d_q} v^d_{qr} z^d_{qr}$. The continuous variable w_a represents the courier flow on arc $a \in A$, while $u^{in}_{(m,t)}$ and $u^{out}_{(m,t)}$ represent the incoming and exiting couriers at period t in zone m. The mathematical formulation is presented as:

$$\begin{aligned} \max \quad & \sum_{d \in D} \mathbb{E}^{rev}_d(z^d_{qr}) - \mathbb{E}^{cost}_{policy}(w_a, u^{in}_{(m,t)}, u^{out}_{(m,t)}) \\ \text{s.t.} \quad & (4.6) - (4.16) \end{aligned} \tag{4.1}$$

The objective function (4.1) calculates the difference between the expected revenue obtained from orders ordered within all districts based on opened restaurants and the expected cost from operating riders. In the remainder of this section, we explain each profit component and the constraints.

Expected revenue. We first define the expected revenue contribution from district d. Each district's revenue depends on the number of arriving requests in all zones covered by the district over the time horizon, $\sum_{m \in M} \sum_{t \in T} b^d_m \lambda_{mt}$, and the expected revenue from customers ordering within district d from Equation (3.5), where total attraction is given by $V^d_q(S^d_q) = v^d_{q0} + \sum_{r \in R^d_q} v^d_{qr} z^d_{qr}$. By multiplying these equations we obtain the total expected revenue from district d over all arriving requests. To incorporate this we change Equation (3.5) to include the model's decision variable z^d_{qr}:

$$\mathbb{E}^{rev}_d(z^d_{qr}) = \frac{\sum_{q \in Q} (v^d_{q0} + \sum_{r \in R^d_q} v^d_{qr} z^d_{qr})^{\gamma^d_q - 1} (\sum_{r \in R^d_q} v^d_{qr} p^d_{qr} z^d_{qr})}{v^d_0 + \sum_{q \in Q} (v^d_{q0} + \sum_{r \in R^d_q} v^d_{qr} z^d_{qr})^{\gamma^d_q}} \cdot \sum_{m \in M} \sum_{t \in T} \lambda_{mt} b^d_m \tag{4.2}$$

By summing over all districts one can obtain the total expected revenue.

Expected costs. The expected rider costs depend on the policy implemented, consisting of the compensation costs and the overhead costs. The compensation costs can be defined similarly as the expected profit for the CB policy, for the FE policy we sum the rider flows over all arcs. This gives us the following:

$$\mathbb{E}^{cost}_{policy}(w_a, u^{in}_{(m,t)}, u^{out}_{(m,t)}) = \hat{c}^{compensation}_{policy} + \hat{c}^{overhead}_{policy} \tag{4.3}$$

Where $\hat{c}^{overhead}_{policy}$ is as defined in subsection 3.7 and $\hat{c}^{compensation}_{policy}$ is defined as:

$$\hat{c}^{compensation}_{policy} = \begin{cases} \frac{\sum_{q \in Q} (v^d_{q0} + \sum_{r \in R^d_q} v^d_{qr} z^d_{qr})^{\gamma^d_q - 1} (\sum_{r \in R^d_q} v^d_{qr} c^d_{qr} z^d_{qr})}{v^d_0 + \sum_{q \in Q} (v^d_{q0} + \sum_{r \in R^d_q} v^d_{qr} z^d_{qr})^{\gamma^d_q}} \cdot \sum_{m \in M} \sum_{t \in T} b^d_m \lambda_{mt} & \text{if policy = CB} \\ \sum_{a \in A} c_a \cdot w_a & \text{if policy = FE} \end{cases} \tag{4.4}$$

Demand satisfaction constraints. To ensure that enough rider capacity is available to fulfill all customer orders within the allowable delivery window ρ, recall that an order place at time t at a restaurant located in zone m by a customer located in zone m' can be picked-up in zone m and delivered in zone m' anywhere within the periods $\{t+\eta ..., t+\rho\}$. The number of related orders was presented by Equation (3.6), and in a similar fashion as for the objective we incorporate decision variable z^d_{qr} to obtain the number of orders, i.e. demand as specified from restaurant to customer, from zone m to m' ordered at period t:

$$\Delta^t_{mm'} = \lambda_{m't} \cdot \frac{\sum_{q\in Q}(v^d_{q0} + \sum_{r\in R^d_q} v^d_{qr} z^d_{qr})^{\gamma^d_q - 1}(\sum_{r\in \hat{R}_m} v^d_{qr} z^d_{qr})}{v^d_0 + \sum_{q\in Q}(v^d_{q0} + \sum_{r\in R^d_q} v^d_{qr} z^d_{qr})^{\gamma^d_q}} \tag{4.5}$$

Recall that the binary parameter b^d_m equals one when zone m is covered by district d and zero otherwise. The number of riders needed to deliver the orders between any two zones over all districts within the delivery window should therefore be at least $\Delta^t_{mm'}$ if m' is covered by d, summed over all districts d, presented in Constraint (4.6). This constraint may include an overlap in demand generated within the delivery window, e.g. when two orders are placed from zone m adjacent to m' at time t_1 and t_2 respectively, and need to be delivered within delivery windows $[t_1, t_2, t_3]$ and $[t_2, t_3, t_4]$ respectively, the constraint will hold if one rider delivers one order from zone m to m' starting at time t_2. Therefore we also need to add a global constraint on total riders versus total demand, presented in Constraint (4.7). Both constraints include larger or equal signs because couriers may also relocate instead of delivering, resulting in larger flow values, but because rider costs are minimized the total flow of couriers will be minimized as well.

$$\sum_{\substack{t'=t+\tau_{mm'} \\ t+\eta\le t'\le t+\rho}} w_{(m,t)(m',t')} \ge \sum_{d\in D} b^d_{m'} \Delta^t_{mm'} \qquad \forall m, m' \in M, \quad t \in T \setminus \{T_{max}-\rho-\eta, ..., T_{max}\} \tag{4.6}$$

$$\sum_{a\in A} w_a \ge \sum_{m\in M}\sum_{m'\in M}\sum_{t\in T}\sum_{d\in D} b^d_{m'} \Delta^t_{mm'} \tag{4.7}$$

$$w_a \in \mathbb{R}_+ \qquad \forall a \in A \tag{4.8}$$

Rider flow constraints. Rider flows are captured via the node-balance constraints and the definitions of $u^{in}_{(m,t)}$ and $u^{out}_{(m,t)}$. For each node $(m,t) \in N$, we require the total incoming flow plus any new riders entering to equal the total outgoing flow plus any riders leaving, presented in Constraint (4.9). Since the total number of riders entering the system must be the same as the total number exiting, we add Constraint (4.10). Finally, we represent operational rules that constrain how long each rider can work on average using the shift duration $[\theta_{min}, \theta_{max}]$ in Constraint (4.11).

$$\sum_{a\in A^-_{(m,t)}} w_a + u^{in}_{(m,t)} = \sum_{a\in A^+_{(m,t)}} w_a + u^{out}_{(m,t)} \qquad \forall (m,t) \in N \tag{4.9}$$

$$\sum_{(m,t)\in N} u^{in}_{(m,t)} = \sum_{(m,t)\in N} u^{out}_{(m,t)} \tag{4.10}$$

$$\theta_{\text{min}} \cdot \sum_{(m,t)\in N} u^{in}_{(m,t)} \le \sum_{a\in A} w_a \le \theta_{\text{max}} \cdot \sum_{(m,t)\in N} u^{in}_{(m,t)} \tag{4.11}$$

$$w_a \in \mathbb{R}_+ \qquad \forall a \in A \tag{4.12}$$

$$u^{in}_{(m,t)} \in \mathbb{R}_+, \quad u^{out}_{(m,t)} \in \mathbb{R}_+ \qquad \forall m \in M, \quad t \in T \tag{4.13}$$

Assortment constraints. Recall that e^r_q is the binary parameter indicating whether restaurant r is of cuisine type q, K^d_q is the minimum number of restaurants of cuisine q to be included in the assortment for service district d, and R^d_q is the set of restaurants that may be included in the assortment for cuisine q in service district d based on the delivery window. Constraint (4.14) enforces that a restaurant r can only be included in the assortment for cuisine q in district d if r indeed belongs to that cuisine q. Constraint (4.15) guarantees that each

cuisine q in district d meets a minimum number of restaurants, providing sufficient diversity to customers.

$$z^d_{qr} \leq e^r_q \qquad \forall r \in R, \quad q \in Q, \quad d \in D \tag{4.14}$$

$$\sum_{r \in R^d_q} z^d_{qr} \geq K^d_q \qquad \forall q \in Q, \quad d \in D \tag{4.15}$$

$$z^d_{qr} \in \{0,1\} \qquad \forall r \in R, \quad q \in Q, \quad d \in D \tag{4.16}$$

The mathematical formulation presented in Equations (4.1) - (4.16) is nonlinear due to the presence of fractional terms, exponents, and products of decision variables. In the following section, we introduce the set partitioning formulation of the problem, in which all feasible assortments are generated a priori. This reformulation eliminates the non-linearity associated with terms involving the exponent γ^d_q, as the base of the exponent becomes a constant. Consequently, this formulation is more suitable for various solution methodologies.

4.2. Enumerated model formulation

Let $\mathcal{B}^d_q \subseteq R^d_q$ represent all the possible assortments for nest q for service district d, complying with any assortment constraints. For each (d,q) pair, we introduce binary decision variable $z_{S^d_q} \in \{0,1\}$ to indicate whether a particular assortment $S^d_q \subseteq \mathcal{B}^d_q$ of restaurants is offered. The continuous variable w_a represents the rider flow on arc $a \in A$, while $u^{in}_{(m,t)}$ and $u^{out}_{(m,t)}$ represent the incoming and exiting riders at node $(m,t) \in N$. The sets and parameters used in the formulation are summarized in Tables 4.1 and 4.2. We first present the mathematical model in its nonlinear form. The linearization of this formulation is presented in Section 4.3. The problem is presented as follows:

$$\begin{aligned} \max \quad & \sum_{d \in D} \mathbb{E}^{rev}_d(z_{S^d_q}) - \mathbb{E}^{cost}_{policy}(w_a, u^{in}_{(m,t)}, u^{out}_{(m,t)}) \\ \text{s.t.} \quad & (4.22) - (4.31) \end{aligned} \tag{4.17}$$

The objective function (4.17) calculates the difference between the expected revenue obtained from orders ordered within all districts based on assortments S^d_q offered and the expected cost from operating riders. In the remainder of this section, we explain each profit component and the constraints, focusing on the changes with respect to the full non-linear model.

Expected revenue. The expected revenue calculation is adjusted by considering the assortments $S^d_q \subseteq R^d_q$ and their related decision variable $z_{S^d_q}$ in Equation (3.5). This results in the following equation:

$$\mathbb{E}^{rev}(z_{S^d_q}) = \frac{\sum_{q \in Q} \sum_{S^d_q \subseteq \mathcal{B}^d_q} V^d_q(S^d_q)^{\gamma^d_q} \pi^d_q(S^d_q) \cdot z_{S^d_q}}{v^d_0 + \sum_{q \in Q} \sum_{S^d_q \subseteq \mathcal{B}^d_q} V^d_q(S^d_q)^{\gamma^d_q} \cdot z_{S^d_q}} \cdot \sum_{m \in M} \sum_{t \in T} b^d_m \lambda_{mt} \tag{4.18}$$

By summing over all districts we obtain the total expected revenue.

Expected costs. The expected rider costs remain largely unchanged, except for the adjustment in the calculation of compensation costs under the CB policy. Specifically, we apply the same transformation used in the expected revenue calculation to derive a revised expression for $\hat{c}^{compensation}_{policy}$, resulting in the following modification:

$$\hat{c}^{compensation}_{policy} = \begin{cases} \frac{\sum_{q \in Q} \sum_{S^d_q \subseteq \mathcal{B}^d_q} V^d_q(S^d_q)^{\gamma^d_q} \sum_{r \in S^d_q} c^d_{qr} \cdot \mathbb{P}^d_{r|q}(S^d_q) \cdot z_{S^d_q}}{v^d_0 + \sum_{q \in Q} \sum_{S^d_q \subseteq \mathcal{B}^d_q} V^d_q(S^d_q)^{\gamma^d_q} \cdot z_{S^d_q}} \cdot \sum_{m \in M} \sum_{t \in T} b^d_m \lambda_{mt} & \text{if policy = CB} \\ \sum_{a \in A} c_a \cdot w_a & \text{if policy = FE} \end{cases} \tag{4.19}$$

Accordingly, the total expected rider costs remain defined as:

$$\mathbb{E}^{cost}_{policy}(w_a, u^{in}_{mt}, u^{out}_{mt}) = \hat{c}^{compensation}_{policy} + \hat{c}^{overhead}_{policy} \tag{4.20}$$

Where $\hat{c}^{overhead}_{policy}$ is as defined in Subsection 3.7.

Demand satisfaction constraints. The demand constraints are adapted to incorporate the same transformation applied in the previous sections. Specifically, we modify the calculation of $\Delta^t_{mm'}$ from Equation (4.5) to reflect the set partitioning formulation. This results in the following revised expression that calculates the number of orders from zone m to m' ordered at period t:

$$\hat{\Delta}^t_{mm'} = \lambda_{m't} \cdot \frac{\sum_{q\in Q}\sum_{S^d_q\subseteq\mathcal{B}^d_q}\sum_{r\in\hat{R}_m\cap S^d_q} v^d_{qr}\cdot V^d_q(S^d_q)^{\gamma^d_q-1}\cdot z^d_{S^d_q}}{v^d_0+\sum_{q\in Q}\sum_{S^d_q\subseteq\mathcal{B}^d_q} V^d_q(S^d_q)^{\gamma^d_q}\cdot z_{S^d_q}} \tag{4.21}$$

While the structure of the constraints remains unchanged, the formulation now incorporates $\hat{\Delta}^t_{mm'}$:

$$\sum_{\substack{t'=t+\tau_{mm'}\\ t+\eta\le t'\le t+\rho}} w_{(m,t)(m',t')} \ge \sum_{d\in D} b^d_{m'}\hat{\Delta}^t_{mm'} \qquad \forall m,m'\in M,\quad t\in T\setminus\{T_{max}-\rho-\eta,...,T_{max}\} \tag{4.22}$$

$$\sum_{a\in A} w_a \ge \sum_{m\in M}\sum_{m'\in M}\sum_{t\in T}\sum_{d\in D} b^d_{m'}\hat{\Delta}^t_{mm'} \tag{4.23}$$

$$w_a\in\mathbb{R}_+ \qquad \forall a\in A \tag{4.24}$$

Rider flow constraints. The rider flow constraints remain unchanged in the set partitioning formulation:

$$\sum_{a\in A^-_{(m,t)}} w_a + u^{in}_{(m,t)} = \sum_{a\in A^+_{(m,t)}} w_a + u^{out}_{(m,t)} \qquad \forall (m,t)\in N \tag{4.25}$$

$$\sum_{(m,t)\in N} u^{in}_{(m,t)} = \sum_{(m,t)\in N} u^{out}_{(m,t)} \tag{4.26}$$

$$\theta_{\text{min}}\cdot\sum_{(m,t)\in N} u^{in}_{(m,t)} \le \sum_{a\in A} w_a \le \theta_{\text{max}}\cdot\sum_{(m,t)\in N} u^{in}_{(m,t)} \tag{4.27}$$

$$w_a\in\mathbb{R}_+ \qquad \forall a\in A \tag{4.28}$$

$$u^{in}_{(m,t)}\in\mathbb{R}_+,\quad u^{out}_{(m,t)}\in\mathbb{R}_+ \qquad \forall m\in M,\quad t\in T \tag{4.29}$$

Assortment constraints. Since all feasible assortments are defined a priori, constraints on the minimum number of required restaurants per assortment S^d_q or cuisine type are directly incorporated into the enumeration of the feasible set $\mathcal{B}^d_q$. As a result, the formulation no longer requires explicit constraints to enforce these conditions, thereby simplifying the model. Constraint (4.30) guarantees that the model selects precisely one subset S^d_q out of all possible subsets of $\mathcal{B}^d_q$ for each (d,q) pair.

$$\sum_{S^d_q\subseteq\mathcal{B}^d_q} z_{S^d_q} = 1 \qquad \forall q\in Q,\quad d\in D \tag{4.30}$$

$$z_{S^d_q}\in\{0,1\} \qquad \forall q\in Q,\quad d\in D,\quad S^d_q\subseteq\mathcal{B}^d_q \tag{4.31}$$

In the following subsection, we discuss the linearization of the above-mentioned nonlinear constraints and the objective function.

4.3. Linearization

The model presented by (4.17)-(4.31) is non-linear due to constraint (4.22), (4.23) and the objective function (4.17). We follow a two-step procedure to transform these expressions into a Mixed-Integer Linear Program (MILP). We first isolate the fractional expressions in the objective and constraints. For district d, let continuous variable y_d capture the fraction

$$y_d = \frac{\sum_{q\in Q}\sum_{S^d_q\subseteq\mathcal{B}^d_q} V^d_q(S^d_q)^{\gamma^d_q}\pi^d_q(S^d_q)\cdot z_{S^d_q}}{v^d_0+\sum_{q\in Q}\sum_{S^d_q\subseteq\mathcal{B}^d_q} V^d_q(S^d_q)^{\gamma^d_q}\cdot z_{S^d_q}} = \frac{f_d}{g_d} \tag{4.32}$$

Similarly, for each zone m and district d, define continuous variable x_{md} to represent the fraction

$$x_{md} = \frac{\sum_{q\in Q}\sum_{S^d_q\subseteq\mathcal{B}^d_q}\sum_{r\in\hat{R}_m\cap S^d_q} v^d_{qr}\cdot V^d_q(S^d_q)^{\gamma^d_q-1}\cdot z^d_{S^d_q}}{v^d_0+\sum_{q\in Q}\sum_{S^d_q\subseteq\mathcal{B}^d_q} V^d_q(S^d_q)^{\gamma^d_q}\cdot z_{S^d_q}} = \frac{h_{md}}{g_d} \tag{4.33}$$

Next we rewrite each fraction as a product of the new variable and a linear function of $z_{S_q^d}$, $g_d y_d = f_d \; \forall d \in D$ and $g_d x_{md} = h_{md} \quad \forall d \in D, \quad m \in M$, and introduce bounds for the newly defined variables. As all the relevant components of the model are larger or equal to zero, the lower bound trivially becomes zero for all variables. For the upper bound we want to find the maximum values these variables can attain. For the upper bound of $y_d \in [0, y_d^U]$ we want to find $y_d^U = \max y_d = \max\{\frac{f_d}{g_d}\}$. f_d can be maximized by noting that $z_{S_q^d} = 1$ for exactly one $S_q^d \;\; \forall q \in Q, \; d \in D$, therefore $\max\{f_d\} = \sum_{q \in Q} \max_{S_q^d \subseteq \mathcal{B}_q^d} V_q^d (S_q^d)^{\gamma_q^d} \pi_q^d(S_q^d)$. Similarly we can show that to minimize g_d we get $\min g_d = v_0^d + \sum_{q \in Q} \min_{S_q^d \subseteq \mathcal{B}_q^d} V_q^d (S_q^d)^{\gamma_q^d}$. Combining these results we get $y_d^U = \frac{\max\limits_z f_d}{\min\limits_z g_d} = \frac{\sum_{q \in Q} \max_{S_q^d \subseteq \mathcal{B}_q^d} V_q^d (S_q^d)^{\gamma_q^d} \pi_q^d(S_q^d)}{v_0^d + \sum_{q \in Q} \min_{S_q^d \subseteq \mathcal{B}_q^d} V_q^d (S_q^d)^{\gamma_q^d}}$. Similarly we find the upper bound for $x_{md} \in [0, x_{md}^U]$. Combining all these results we obtain the following formulation:

$$\max \quad \sum_{d \in D} y_d \cdot \sum_{m \in M} \sum_{t \in T} b_m^d \lambda_{mt} - \mathbb{E}_{policy}^{cost}(w_a, u_{mt}^{in}, u_{mt}^{out}) \tag{4.34}$$

s.t. (4.24) – (4.31)

$$g_d y_d = f_d \qquad \forall d \in D \tag{4.35}$$

$$g_d x_{md} = h_{md} \qquad \forall d \in D, \quad m \in M \tag{4.36}$$

$$\sum_{\substack{t' = t + \tau_{mm'} \\ t+\eta \le t' \le t+\rho}} w_{(m,t)(m',t')} \ge \lambda_{m't} \sum_{d \in D} b_{m'}^d x_{md} \tag{4.37}$$

$$\forall m, m' \in M, \; t \in T \setminus \{T_{max} - \rho - \eta, ..., T_{max}\}$$

$$\sum_{a \in A} w_a \ge \sum_{m \in M} \sum_{m' \in M} \sum_{t \in T} \lambda_{m't} \sum_{d \in D} b_{m'}^d x_{md} \tag{4.38}$$

$$y_d, f_d, g_d \in [0, y_d^U], \quad x_{md}, h_{md} \in [0, x_{md}^U] \qquad \forall d \in D, \quad m \in M \tag{4.39}$$

Note that for the CB policy we can similarly reformulate the fractional term. Yet still, constraints (4.35) and (4.36) are bilinear in terms of decision variables $z_{S_q^d}$, y_d, g_d and x_{md}. We can further linearize these constraints by applying the commonly used linearization technique by Charnes and Cooper [1973], resulting in adding additional constraints to the model. We introduce auxiliary continuous variables $l_{S_q^d}$ and $k_{S_q^d}^m$ such that $l_{S_q^d} = z_{S_q^d} \cdot y_d$ and $k_{S_q^d}^m = z_{S_q^d} \cdot x_{md}$. We then add the following linearization constraints:

$$l_{S_q^d} \le y_d \qquad \forall q \in Q, \quad d \in D, \quad S_q^d \subseteq \mathcal{B}_q^d \tag{4.40}$$

$$l_{S_q^d} \le y_d^U \cdot z_{S_q^d} \qquad \forall q \in Q, \quad d \in D, \quad S_q^d \subseteq \mathcal{B}_q^d \tag{4.41}$$

$$l_{S_q^d} \ge y_d - y_d^U \cdot (1 - z_{S_q^d}) \qquad \forall q \in Q, \quad d \in D, \quad S_q^d \subseteq \mathcal{B}_q^d \tag{4.42}$$

$$k_{S_q^d}^m \le x_{md} \qquad \forall q \in Q, \quad d \in D, \quad S_q^d \subseteq \mathcal{B}_q^d, \quad m \in M \tag{4.43}$$

$$k_{S_q^d}^m \le x_{md}^U \cdot z_{S_q^d} \qquad \forall q \in Q, \quad d \in D, \quad S_q^d \subseteq \mathcal{B}_q^d, \quad m \in M \tag{4.44}$$

$$k_{S_q^d}^m \ge x_{md} - x_{md}^U \cdot (1 - z_{S_q^d}) \qquad \forall q \in Q, \quad d \in D, \quad S_q^d \subseteq \mathcal{B}_q^d, \quad m \in M \tag{4.45}$$

$$l_{S_q^d} \in [0, y_d^U], \quad k_{S_q^d}^m \in [0, x_{md}^U] \qquad \forall q \in Q, \quad d \in D, \quad S_q^d \subseteq \mathcal{B}_q^d, \quad m \in M \tag{4.46}$$

By incorporating the new auxiliary variables into the formulation the bilinear constraints (4.35) and (4.36) respectively change to:

$$\sum_{q \in Q} \sum_{S_q^d \subseteq \mathcal{B}_q^d} V_q^d (S_q^d)^{\gamma_q^d} \pi_q^d(S_q^d) \cdot z_{S_q^d} = v_0^d \cdot y_d + \sum_{q \in Q} \sum_{S_q^d \subseteq \mathcal{B}_q^d} V_q^d (S_q^d)^{\gamma_q^d} \cdot l_{S_q^d} \qquad \forall d \in D \tag{4.47}$$

$$\sum_{q \in Q} \sum_{S_q^d \subseteq \mathcal{B}_q^d} \sum_{r \in \hat{R}_m \cap S_q^d} v_{qr}^d \cdot V_q^d (S_q^d)^{\gamma_q^d - 1} \cdot z_{S_q^d} = v_0^d \cdot x_{md} + \sum_{q \in Q} \sum_{S_q^d \subseteq \mathcal{B}_q^d} V_q^d (S_q^d)^{\gamma_q^d} \cdot k_{S_q^d}^m \quad \forall d \in D, \quad m \in M \tag{4.48}$$

These linearizations replace all the bilinear terms the model, allowing it to be formulated as a MILP. To show the equivalence between the original non-linear model $\mathcal{M}_{NL}$ and the enumerated linearized model $\mathcal{M}_{EL}$, we

prove the following theorem:

Theorem 4.3.1 (Equivalence between $\mathcal{M}_{NL}$ and $\mathcal{M}_{EL}$)**.** *Let $\mathcal{M}_{NL}$ be the original non-linear optimization model, defined over decision variables $(z^d_{qr}, w_a, u^{in}_{(m,t)}, u^{out}_{(m,t)})$, with an objective function and constraints that are non-linear due to fractional expressions involving binary variables z^d_{qr}. Let $\mathcal{M}_{EL}$ be the enumerated and linearized model derived from $\mathcal{M}_{NL}$ by enumerating all feasible subsets $S^d_q \in \mathcal{B}^d_q$ that represent the possible assortments for each cuisine q in district d, thus replacing z^d_{qr} with $z_{S^d_q}$, and introducing auxiliary variables and additional constraints to linearize all fractional and non-linear terms, resulting in a Mixed-Integer Linear Program (MILP). Then, $\mathcal{M}_{NL}$ and $\mathcal{M}_{EL}$ are equivalent in terms of feasible solutions and optimal objective value. In particular:*

(i) For every feasible solution of $\mathcal{M}_{NL}$, there exists a corresponding feasible solution of $\mathcal{M}_{EL}$ that attains the same objective value.

(ii) For every feasible solution of $\mathcal{M}_{EL}$, there exists a corresponding feasible solution of $\mathcal{M}_{NL}$ that attains the same objective value.

As a consequence, any globally optimal solution to the enumerated and linearized model $\mathcal{M}_{EL}$ corresponds to a globally optimal solution of the original non-linear model $\mathcal{M}_{NL}$.

Proof. The original model $\mathcal{M}_{NL}$ uses binary variables z^d_{qr} to implicitly determine the assortment of restaurants selected for each cuisine q in district d. These z^d_{qr} variables must satisfy constraints such as minimum number of selected restaurants per cuisine and other feasibility conditions related to service and routing. In the enumerated model $\mathcal{M}_{EL}$, we explicitly list all feasible subsets $\mathcal{B}^d_q$ of restaurants for each cuisine q and district d. Each feasible subset $S^d_q \in \mathcal{B}^d_q$ corresponds to a unique pattern of z^d_{qr} variables being 1 for $r \in S^d_q$ and 0 otherwise. Thus, selecting $z_{S^d_q} = 1$ in $\mathcal{M}_{EL}$ replicates exactly one feasible configuration of z^d_{qr} in $\mathcal{M}_{NL}$.

Conversely, any feasible assignment of $\{z^d_{qr}\}$ in $\mathcal{M}_{NL}$ that satisfies the assortment constraints defines a unique subset S^d_q, which is included in $\mathcal{B}^d_q$ by construction. Setting $z_{S^d_q} = 1$ and $z_{S'^d_q} = 0$ for $S'^d_q \neq S^d_q$ in $\mathcal{M}_{EL}$ reproduces this configuration. Thus, there is a one-to-one correspondence between feasible assortment selections in the original and enumerated models. The remaining variables $(w_a, u^{in}_{(m,t)}, u^{out}_{(m,t)})$ and their constraints are preserved in the linearization step. Hence, any feasible solution $(z^d_{qr}, w_a, u^{in}_{(m,t)}, u^{out}_{(m,t)})$ in $\mathcal{M}_{NL}$ maps to at least one $(z_{S^d_q}, w_a, u^{in}_{(m,t)}, u^{out}_{(m,t)}, ...)$ solution in $\mathcal{M}_{EL}$, and vice versa.

The original model $\mathcal{M}_{NL}$ contains fractional terms in the objective and in some constraints. After enumeration, some non-linear terms become constants associated with each assortment S^d_q, resulting in the following simplified form of non-linearity:

$$\sum_{d \in D} \frac{f_d(\mathbf{z})}{g_d(\mathbf{z})}$$

Both functions in the numerator and denominator of are linear in $z_{S^d_q}$. Similarly the non-linear constraints can be written in this form. This makes the problem a Mixed-Integer Linear Fractional Program (MILFP). By introducing auxiliary variables and linearization constraints we apply the Charnes-Cooper transformation [Charnes and Cooper, 1973], reformulating the MILFP using bilinear constraints, and by using Glover's linearization turning it into the equivalent MILP, obtaining a set of linear equalities and inequalities that exactly replicate the fractional relationships. This linearization is done fraction-by-fraction and since linear constraints are closed under addition, summing multiple linearized fraction does not break any equivalence. This means that when summing over all districts $d \in D$ in the objective there is no loss of optimality in summing the individually linearized terms. Therefore, the result is not an approximation but an exact reformulation: for any feasible assignment of $(z_{S^d_q}, w_a, u^{in}_{(m,t)}, u^{out}_{(m,t)})$, the auxiliary variables can be chosen to match the fractional value from $\mathcal{M}_{NL}$ exactly. Hence, the objective value obtained in $\mathcal{M}_{EL}$ for any given feasible solution matches exactly the objective value of the corresponding solution in $\mathcal{M}_{NL}$.

Since there is a bijection between feasible solutions of $\mathcal{M}_{NL}$ and $\mathcal{M}_{EL}$ and their respective objective values are identical, optimal solutions map between the two models. If $(z^*_{S^d_q}, w^*_a, u^{in^*}_{(m,t)}, u^{out^*}_{(m,t)}, \dots)$ is optimal in $\mathcal{M}_{EL}$ with optimal value Z^*, then the corresponding solution in $\mathcal{M}_{NL}$ attains at least Z^*. If there were a better solution in $\mathcal{M}_{NL}$, it would map to a solution in $\mathcal{M}_{EL}$ with a better objective, contradicting the optimality in $\mathcal{M}_{EL}$. □

In this regard, the original model is reformulated to the enumerated model and transformed into a MILP, which can effectively be solved by a typical branch-and-bound solver like Gurobi, able of reaching global optimality. As shown by the theorem above, this ensures we have also found the global optimum of the original non-linear model $\mathcal{M}_{NL}$. However, the problem grows exponentially in variables due to the number of possible combinations of restaurants for the assortments. Therefore, we need to carefully search the solution space for which we present our resolution approach in Chapter 5.

5
Resolution approaches

This chapter explores the methodologies employed to solve the RSRDP. We present an overview of resolution techniques ranging from exact algorithms to heuristic methods, each tailored to address the trade-off between computational efficiency and solution quality. Furthermore, a benchmark solution method for a simplified separated model provides a baseline for evaluating the added value of the integrated optimization. We aim to provide a comprehensive narrative of different resolution approaches, guiding the reader through our decisions and discoveries.

Each method offers unique strengths and limitations, contributing to a holistic understanding of the problem. Exact methods, including using a standard solver and a Benders decomposition framework, resulted computationally intensive. Therefore, heuristic methods are explored. We present our tailored Iterative Assortment Generation (IAG) algorithm that strikes a balance between computational feasibility and solution quality, leveraging problem-specific insights to navigate the solution space efficiently. Finally, column generation provided critical insights that shaped the development of the heuristic algorithm. We discuss the challenges faced when using column generation techniques and justify its exclusion from our main results. The proposed methodologies are evaluated through numerical experiments, comparing the useful algorithms to assess their performance in terms of solution quality and computational efficiency, which are presented in Chapter 6.

5.1. Benchmark: separated assortment and rider optimization

To establish a baseline for evaluating the benefits of integrated optimization, we adopt a two-stage benchmark. In the first stage, assortment offerings are optimized independently for each service district, without considering rider decisions. In the second stage, the rider dimensioning problem is solved with the previously chosen assortments treated as fixed. This approach mirrors a straightforward industry practice in which restaurant choices (restaurant assortments) and rider operations are managed as separate processes. Below, we describe how we obtain a polynomial-time solution for assortment selection using a method inspired by Davis et al. [2014], and then integrate these results into the rider dimensioning problem.

Assortment model. In this approach, each service district determines the optimal restaurants to offer for each $(q, d),\ q \in Q,\ d \in D$ pair. Under the assumptions $v^d_{q0} = 0$ and $\gamma^d_q \in [0, 1]\ \forall q \in Q,\ d \in D$, Davis et al. [2014] show that for a single nest, considering only nested-by-revenue assortments is optimal. We extend this result to all districts by noting that each district's assortment problem is independent when rider dimensioning is not considered.

To determine NBR assortments, for every service district $d \in D$, and restaurants in each nest $q \in Q$ are ranked in descending order of their individual revenue contributions. A candidate assortment is formed by adding restaurants one at a time, where the process stops when adding any additional restaurant would reduce the marginal increase in expected revenue to zero or negative. We denote the set of NBR assortments for each cuisine-district pair (q, d) by Ψ^d_q. To find the NBR assortments for every service district we extend the formulation by Davis et al. [2014] to include multiple assortment decisions. We define decision variables $\chi_d \in \mathbb{R}_+$ and

$\psi_q^d \in \mathbb{R}_+$ such that $\psi_q^d = \max_{S_q^d \subseteq R_q^d} V_q^d(S_q^d)^{\gamma_q^d}(\pi_q^d(S_q^d) - \chi_d)$. The linear program is then defined as:

$$\mathcal{R}^* = \min \sum_{d\in D} \chi_d \tag{5.1}$$

$$\text{s.t.} \quad v_0^d \chi_d \geq \sum_{q\in Q} \psi_q^d \qquad \forall d \in D \tag{5.2}$$

$$\psi_q^d \geq V_q^d(S_q^d)^{\gamma_q^d}\Big(\sum_{m\in M}\sum_{t\in T} b_m^d \lambda_{mt} \pi_q^d(S_q^d) - \chi_d\Big) \quad \forall S_q^d \subseteq \Psi_q^d, \quad q \in Q, \quad d \in D \tag{5.3}$$

$$\chi_d \in \mathbb{R}_+,\ \psi_q^d \in \mathbb{R}_+ \qquad \forall q \in Q, \quad d \in D \tag{5.4}$$

This program provides the optimal NBR assortments and is polynomial solvable as it has $|D|(1 + |Q|)$ decision variables and $|D|(1 + |Q|(1 + |R_d|))$ constraints.

Rider dimensioning. Once the optimal restaurants have been chosen for each nest q and district d, we fix these assortment decisions by setting $z_{S_q^d} = 1$ for the chosen NBR assortments and zero otherwise. We then proceed to solve the problem with the fixed $z_{S_q^d}$ variables, providing us with the total expected profit of the benchmark solution. This is still computed as the difference between the revenues from the selected assortments and the rider costs related to the applied policy. Because the two stages are decoupled, this approach provides a baseline to compare the integrated model to. Additionally, the NBR assortments are used as an initialization for the resolution approaches presented in the next sections.

5.2. Exact solution method

Exact algorithms guarantee proven optimality for the underlying problem but can be computationally demanding. In this section, we detail how one such approach was applied to the linearized and enumerated RSRDP model from Chapter 4. Specifically, we explore the impact of Benders decomposition, and propose a column generation framework to solve the problem, including the incorporation of Benders cuts to speed up convergence.

5.2.1. Benders decomposition

Benders decomposition is employed to decompose the problem and solve more efficiently: the master problem (MP) on assortment selection to maximize revenue and the rider subproblem (SP) to minimize rider costs. In our formulation, the master problem is an integer program that selects an assortment for each service district and cuisine nest, while the subproblem is a continuous optimization model that determines the allocation of riders throughout the system based on the chosen assortments. Mathematically, for the integrated problem formulated in Chapter 4, the Benders decomposition formulation is the following:

Master Problem (MP).

$$\max \sum_{d\in D} y_d \cdot \sum_{m\in M}\sum_{t\in T} b_m^d \lambda_{mt} - \Theta \tag{5.5}$$

$$\text{s.t.} \sum_{S_q^d \in \mathcal{B}_q^d} z_{S_q^d} = 1 \qquad \forall q \in Q, \quad d \in D \tag{5.6}$$

(4.40) – (4.48)

$$\Theta \geq Z + \sum_{d\in D}\sum_{m\in M} \phi_{md} \cdot (x_{md} - \hat{x}_{md}) \tag{5.7}$$

Constraint (5.7) represents the Bender's cut from the subproblem, where ϕ_{md} is the dual variable of constraint (5.11) in the subproblem. Given a solution $\hat{x}_{md}$ from the MP, the subproblem is presented.

Subproblem (SP).

$$\min \quad Z = \mathbb{E}^{cost}_{policy}(w_a, u^{in}_{(m,t)}, u^{out}_{(m,t)}) \tag{5.8}$$

s.t. (4.25) – (4.29)

$$\sum_{\substack{t'=t+\tau_{mm'} \\ t+\eta \le t' \le t+\rho}} w_{(m,t)(m',t')} \ge \lambda_{m't} \sum_{d \in D} b^d_{m'} x_{md} \quad \forall m, m' \in M,\ t \in T \setminus \{T_{max} - \rho - \eta, ..., T_{max}\} \tag{5.9}$$

$$\sum_{a \in A} w_a \ge \sum_{m \in M} \sum_{m' \in M} \sum_{t \in T} \lambda_{m't} \sum_{d \in D} b^d_{m'} x_{md} \tag{5.10}$$

$$x_{md} = \hat{x}^*_{md} \qquad \forall m \in M,\ d \in D \tag{5.11}$$

The optimal vector $\hat{x}_{md}$ represents the demand generated by the assortment plan, is obtained by solving the master problem. The resulting cut from the SP is subsequently added to the MP, which is a process that repeats itself until the MP reaches optimality. By incorporating Benders optimality cuts into the MP, the formulation is progressively tightened, enhancing convergence to the optimal solution.

This approach allows for a comparison of the computational performance between a standard solver and one incorporating Benders decomposition. However, since both methods involve the enumeration of all possible assortments, they remain intractable for larger instances. To address this, we explored an exact column generation (CG) algorithm combined with a branch-and-price method, incorporating Benders cuts to expedite convergence at each node in the branch-and-bound tree, explained in more detail in the next subsection.

5.2.2. Column generation

Column generation (CG) is an iterative method that helps deal with very large sets of possible solutions, called columns in the optimization model, by introducing only a few columns at a time, rather than enumerating them all at once. The idea is to start with a Restricted Master Problem (RMP) that uses only a small subset of columns (assortments in our case). We solve this smaller problem, then look for additional columns using the Pricing Problem (PP) that could improve the current solution. If we find any, we add them to the RMP and resolve. This process repeats until no further improvements are possible, guaranteeing an optimal solution to the relaxed problem.

Initially, we believed that the ideal assortments might be close to the NBR assortments because our preliminary results suggested these assortments were strong candidates. Column generation seemed like a good fit: it could begin with these NBR assortments and then incrementally add only new assortments that would truly enhance the objective, thereby handling the otherwise massive variety of possible choices.

However, despite its elegance, our investigations revealed that the associated PP became numerically unstable and intractable due to non-convex terms. We used the `Gurobi 11.0.1` solver, which was unable to obtain the optimal solution for the PP. As the scale of the instance and the diversity of restaurants grew, the combinatorial explosion in the PP caused severe computational bottlenecks. Repeatedly, the solver would generate columns that were effectively duplicates, mainly due to large dual variable magnitudes that skewed the reduced-cost calculations. In this section, we discuss the intractability of the proposed column generation approach for our assortment optimization component, how a Benders decomposition was envisioned to handle additional routing decisions, and what the branch-and-price framework would look like to obtain an optimal solution to the problem if the PP is solvable. Potential strategies for tailoring the method to make it computationally tractable on large-scale instances is presented using our current knowledge, providing possible insights for future research directions and the inspiration for our Iterative Assortment Generation heuristic presented in the next section.

Column generation formulation

Our approach initializes with a small subset of columns (assortments), yielding an Restricted Master Problem (RMP) where constraints (5.13)–(5.24) are enforced over a subset of potential assortments $\hat{\mathcal{B}}^d_q$. We solve this RMP to optimality in a relaxed setting and then solve a Pricing Problem (PP) to identify assortments with positive reduced cost. The highest newly identified columns are added to the RMP, and the process repeats until no more improving columns exist. Note that this includes only the assortment optimization, we were planning on adding the expected costs from the riders later using the Benders decomposition framework presented in the previous subsection. The RMP is defined as:

Restricted Master Problem (RMP).

$$\max \quad \sum_{d\in D} y_d \cdot \sum_{m\in M}\sum_{t\in T} b_m^d \lambda_{mt} \tag{5.12}$$

$$\text{s.t.} \quad \sum_{S_q^d \in \hat{\mathcal{B}}_q^d} z_{S_q^d} = 1 \qquad \forall q \in Q, \quad d \in D \tag{5.13}$$

$$\sum_{q\in Q}\sum_{S_q^d\subseteq\hat{\mathcal{B}}_q^d} V_q^d(S_q^d)^{\gamma_q^d}\pi_q^d(S_q^d)\cdot z_{S_q^d} = v_0^d \cdot y_d + \sum_{q\in Q}\sum_{S_q^d\subseteq\hat{\mathcal{B}}_q^d} V_q^d(S_q^d)^{\gamma_q^d}\cdot l_{S_q^d} \qquad \forall d \in D \tag{5.14}$$

$$\sum_{q\in Q}\sum_{S_q^d\subseteq\hat{\mathcal{B}}_q^d}\sum_{r\in\hat{R}_m\cap S_q^d} v_{qr}^d \cdot V_q^d(S_q^d)^{\gamma_q^d-1}\cdot z_{S_q^d} = v_0^d x_{md} + \sum_{q\in Q}\sum_{S_q^d\subseteq\hat{\mathcal{B}}_q^d} V_q^d(S_q^d)^{\gamma_q^d}\cdot k_{S_q^d}^m \quad \forall d \in D, \quad m \in M \tag{5.15}$$

$$l_{S_q^d} \le y_d \qquad \forall q \in Q, \quad d \in D, \quad S_q^d \subseteq \hat{\mathcal{B}}_q^d \tag{5.16}$$

$$l_{S_q^d} \le y_d^U \cdot z_{S_q^d} \qquad \forall q \in Q, \quad d \in D, \quad S_q^d \subseteq \hat{\mathcal{B}}_q^d \tag{5.17}$$

$$l_{S_q^d} \ge y_d - y_d^U \cdot (1 - z_{S_q^d}) \qquad \forall q \in Q, \quad d \in D, \quad S_q^d \subseteq \hat{\mathcal{B}}_q^d \tag{5.18}$$

$$k_{S_q^d}^m \le x_{md} \qquad \forall q \in Q, \quad d \in D, \quad S_q^d \subseteq \hat{\mathcal{B}}_q^d, \quad m \in M \tag{5.19}$$

$$k_{S_q^d}^m \le x_{md}^U \cdot z_{S_q^d} \qquad \forall q \in Q, \quad d \in D, \quad S_q^d \subseteq \hat{\mathcal{B}}_q^d, \quad m \in M \tag{5.20}$$

$$k_{S_q^d}^m \ge x_{md} - x_{md}^U \cdot (1 - z_{S_q^d}) \qquad \forall q \in Q, \quad d \in D, \quad S_q^d \subseteq \hat{\mathcal{B}}_q^d, \quad m \in M \tag{5.21}$$

$$z_{S_q^d} \in [0,1] \quad (relaxed) \qquad \forall q \in Q,\ d \in D,\ S_q^d \in \hat{\mathcal{B}}_q^d \tag{5.22}$$

$$l_{S_q^d} \in [0, y_d^U], \quad k_{S_q^d}^m \in [0, x_{md}^U] \qquad \forall q \in Q,\ d \in D,\ S_q^d \in \hat{\mathcal{B}}_q^d,\ m \in M \tag{5.23}$$

$$y_d \in [y_d^L, y_d^U], \quad x_{md} \in [x_{md}^L, x_{md}^U] \qquad \forall d \in D,\ m \in M \tag{5.24}$$

Derivation of the Pricing Problem (PP). In the column generation framework, the Pricing Problem is derived from the RMP by exploiting its dual solution. After solving the RMP, each constraint is associated with a dual variable. These dual values represent the marginal worth of relaxing the corresponding RMP constraints. To determine if a new column (assortment) could improve the current solution, we compute its reduced costs, that is, the net benefit of introducing that column into the master problem. Mathematically, for a candidate column defined by binary variables $z_{qr}^d = 1$ if restaurant r is included in the newly generated assortment S_q^d for cuisine q in district d, and $z_{qr}^d = 0$ otherwise, the reduced cost depends on dual variables of the constraints. Let ϕ_q^d represent the dual variable for constraint (5.13), σ_d for constraint (5.14), μ_m^d for constraint (5.15), $\alpha_{q,d,S_q^d}^{(1)}$ and $\alpha_{q,d,S_q^d}^{(2)}$ for constraints (5.17) and (5.18) respectively, and $\beta_{d,m,q,S_q^d}^{(1)}$ and $\beta_{d,m,q,S_q^d}^{(2)}$ for constraints (5.20) and (5.21) respectively. If R_q^d is the set of restaurants available for cuisine q in district d, the reduced cost $\bar{c}_q^d(z_{qr}^d)$ can be formulated as:

$$\begin{aligned}\bar{c}_q^d(z_{qr}^d) = &- \phi_q^d - \sigma_d \left(v_{q0}^d + \sum_{r\in R_q^d} v_{qr}^d z_{qr}^d\right)^{\gamma_q^d} \left(\frac{\sum_{r\in R_q^d} p_{qr}^d v_{qr}^d z_{qr}^d}{v_{q0}^d + \sum_{r\in R_q^d} v_{qr}^d z_{qr}^d}\right)\left(\sum_{m\in M}\sum_{t\in T} b_m^d \lambda_{mt}\right) \\ &- \sum_{m\in M}\mu_m^d \sum_{r\in\hat{R}_m\cap R_q^d} v_{qr}^d z_{qr}^d \left(v_{0q}^d + \sum_{r\in R_q^d} v_{qr}^d z_{qr}^d\right)^{\gamma_q^d-1} + \sum_{S_q^d\in\hat{\mathcal{B}}_q^d}(\alpha_{q,d,S_q^d}^{(1)} - \alpha_{q,d,S_q^d}^{(2)}) y_d^U \\ &+ \sum_{m\in M}\sum_{S_q^d\in\hat{\mathcal{B}}_q^d} b_m^d \left(\beta_{d,m,q,S_q^d}^{(1)} - \beta_{d,m,q,S_q^d}^{(2)}\right) x_{md}^U\end{aligned} \tag{5.25}$$

This expression measures the difference between the column's potential contribution to the objective and the cost of its impact on the RMP constraints (weighted by the duals). The PP then seeks the assortment, the vector z_{qr}^d that maximizes $\bar{c}_q^d(z_{qr}^d)$, subject to feasibility constraints. In compact form, the PP is then presented as:

Pricing Problem (PP).

$$\max \quad \bar{c}_q^d(z_{qr}^d) \tag{5.26}$$

$$\text{s.t.} \quad \sum_{r \in R_q^d} z_{qr}^d \leq |R_q^d| \qquad \text{(trivial upper bound)} \quad \forall q \in Q, \quad d \in D \tag{5.27}$$

$$\sum_{r \in R_q^d} z_{qr}^d \geq K_q^d \qquad \text{(if minimum selection constraints apply)} \quad \forall q \in Q, \quad d \in D \tag{5.28}$$

$$z_{qr}^d \in \{0,1\} \qquad \forall q \in Q, \quad d \in D, \quad r \in R_q^d \tag{5.29}$$

Where K_q^d is the optional lower bound on the assortment size. The column z_{qr}^d with the highest $\bar{c}_q^d(z_{qr}^d) > 0$ is added to the RMP. This iterative loop continues until no improving columns exist, guaranteeing optimality in the continuous relaxation of the RMP. However, a challenge emerges in our approach because the resulting PP is non-convex. The non-convexity arises from several sources: non-linear exponents, fractional expressions and multiplications of dual variables. The combination of these factors means that the PP's objective function is highly non-convex with respect to the binary decision variables z_{qr}^d.

Our implementation of the PP uses `Gurobi 11.0.1`, which is capable of handling certain classes of non-convex optimization. Because the PP incorporates binary variables, we initially expected that Gurobi might effectively tackle the problem despite its non-convex components. During testing, the solver did generate promising assortments at early stages. However, we observed numerical instabilities that caused repeated generation of the same columns. In particular, the large magnitudes of some dual variables led to unstable reduced-cost calculations, causing the solver to repeatedly identify columns that were essentially duplicates. As we increased the potential restaurant pool, the combinatorial complexity of the PP grew accordingly. This growth inevitably slowed the solver, further compounding numerical precision issues. From our experiments, it appears that reliably solving this non-convex PP for large instances would likely require more advanced techniques in non-convex optimization, techniques that are beyond the current practical scope. As a result, a heuristic method tailored to this particular pricing structure may be more viable for large-scale settings. Appendix C demonstrates how the riders component can be merged into the column generation procedure by applying a full branch-and-price framework with Benders decomposition where column generation is invoked at each node of the search tree.

Although the CG exploration proved challenging, it provided valuable insights that informed the decision to pivot towards heuristic approaches for solving larger instances. This exploration serves as a foundation for the next section, where we introduce our heuristic algorithm that can handle more complex scenarios.

5.3. Heuristic solution method

Given the complexity and scale of RSRDP, heuristic methods present a practical alternative to exact algorithms. While exact methods offer optimality, they struggle to scale for real-world scenarios involving hundreds of available restaurants. In contrast, heuristics provide near-optimal solutions within reasonable computational times by leveraging problem-specific insights to guide the search. We exploit observed solution space properties, such as the dominance of nested-by-revenue assortments, to design more efficient algorithms. The primary challenge lies in the exponential growth of potential assortments as the number of available restaurants increases. To address this, we focus on selecting a subset of assortments rather than enumerating all possibilities. Using these problem-specific insights, we propose the Iterative Assortment Generation (IAG) algorithm, which incrementally generates new assortments to include in the set of potential options. This section first introduces the tailored IAG algorithm. We then present an extension inspired by the scoring system of the Adaptive Large Neighborhood Search (ALNS) algorithm, providing additional insights into the mechanics and performance of the IAG approach.

5.3.1. Iterative Assortment Generation

In addressing the problem, we propose a novel solution methodology defined as the Iterative Assortment Generation (IAG) algorithm that incorporates Benders decomposition. This approach is designed to balance computational efficiency with solution quality, and it leverages problem-specific insights to navigate the vast combinatorial space of potential restaurant assortments. Figure 5.1 summarizes the overall methodology. The IAG algorithm starts with decomposing the problem into a master and subproblem as described in subsection 5.2.1. Next, the initial set of assortments that form the Restricted Master Problem (RMP) is generated, which are the

nested-by-revenue (NBR) assortments described in subsection 5.1 and detailed again below. The RMP is solved using a standard solver such as `Gurobi`. The routing subproblem is solved to generate dual information that produces Benders cuts, which are integrated back into the RMP. New assortments are generated based on the selected assortments from the current iteration, expanding the solution space. To maintain computational efficiency, assortments that are unused over multiple iterations are removed from the restricted problem based on an inactivity threshold. The process iterates until a stopping criterion is met, which we define as a time limit or maximum iteration count.

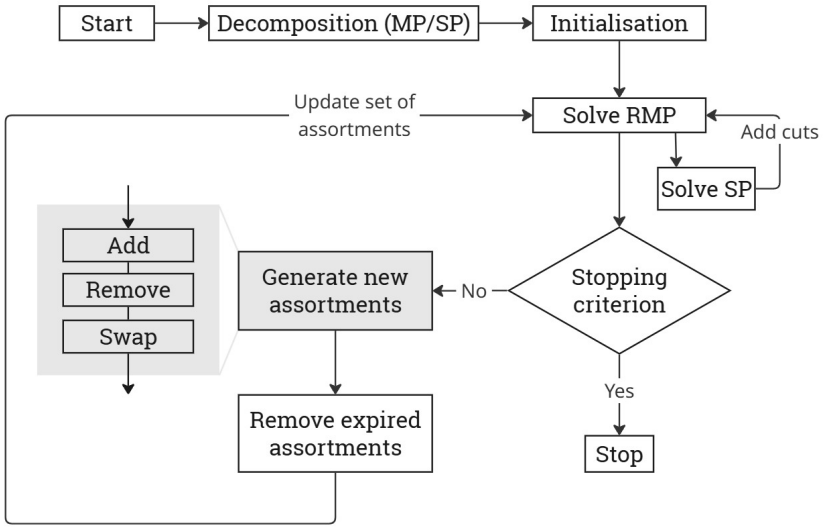

Figure 5.1: Overview diagram of the IAG algorithm.

Our initial exploration focused on an exact solution framework employing column generation combined with a branch-and-price method augmented with Benders decomposition. In this exact framework, the master problem contains the assortment selection variables and is iteratively tightened by incorporating Benders cuts derived from the routing subproblem. Unfortunately, the non-convexity of the associated pricing problem led to intractability. To address this, we develop the Iterative Assortment Generation (IAG) algorithm described in Algorithm 1, which is a heuristic that leverages the structure of the problem without relying on dual-based pricing.

At the heart of the IAG algorithm are three heuristic operators (`add`, `remove`, and `swap`) that generate new assortments by exploring the neighborhood of the current best solution. These operators function similarly to the pricing step in column generation; however, they are tailored to our problem structure and do not depend on dual variables. This allows us to maintain integrality throughout the iterative process, ensuring that every generated solution is feasible in the original problem space. Algorithm 1 presents the pseudo-code of the IAG algorithm. Each component of the algorithm is explained in more detail in the subsequent subsections.

Algorithm 1 Iterative Assortment Generation Algorithm

Input: Data instance, time limit T_{limit}, maximum number of iterations I_{limit}, inactivity threshold φ, last used iteration $\mathcal{I}_{\text{last}}$.
Output: Best objective P^* and associated assortments $\mathcal{A}^*$.
Initialize: $P^* \leftarrow$ NBR objective, $\mathcal{A}^* \leftarrow$ NBR assortments, iteration counter $i \leftarrow 0$, $\mathcal{I}_{\text{last}}(A^*) \leftarrow 0$.
while $i < I_{\text{limit}}$ ***and*** *runtime* $< T_{\text{limit}}$ **do**
  Solve the **Restricted Master Problem (RMP)** using the Benders decomposition method
  Extract selected assortments $\mathcal{A}_i$ and objective P_i
  if $P_i > P^*$ **then**
    Update $P^* \leftarrow P_i, \mathcal{A}^* \leftarrow \mathcal{A}_i$
  Update $\mathcal{I}_{\text{last}}(A^*) \leftarrow i$
  Generate new assortments based on selected assortments and add to RP `// Section: generation`
  Remove unused assortments based on inactivity threshold and remove from RP `// Section: removal`
  Increment iteration counter $i \leftarrow i + 1$
return P^* *and* $\mathcal{A}^*$

For small problem instances, we can validate the heuristic's performance by comparing its results to the exact

optimal solution obtained using the Benders decomposition method described earlier. By pre-computing the optimal solution, we can introduce a condition in the algorithm's while loop to terminate when the optimal solution is reached. This allows us to measure the number of iterations and the computational time required for the algorithm to find the optimal solution, provided it succeeds. To implement this, we introduce a small tolerance ϵ and incorporate a check within the while loop. The loop is broken when $|P_i - P^*| < \epsilon$, indicating that the algorithm has successfully identified the optimal solution.

Initial Nested-By-Revenue assortments.

Under the assumptions that the within-nest no-purchase option satisfies $v_{q0}^d = 0$ and that the dissimilarity parameter $\gamma_q^d \leq 1$ for all $d \in D,\ q \in Q$, Davis et al. [2014] showed that the optimal assortment can be constructed by considering only the nested-by-revenue (NBR) assortments. For each cuisine nest, restaurants are indexed in descending order of revenue, and the optimal assortment is obtained by including restaurants sequentially until the marginal revenue becomes non-positive. These NBR solutions for each district-cuisine pair (d, q) are used as the initial candidates in our IAG algorithm.

Assortment generation.

The pseudo-code for generating new assortments in the IAG framework is presented in Algorithm 2. It focuses on refining the set of possible assortments ($\mathcal{A}_{poss}$) by leveraging modifications to previously optimal selected assortments ($\mathcal{A}_{prev}$). At each iteration, the algorithm evaluates restaurants within the previously optimal assortment for a specific cuisine and district combination (d, q) ($\mathcal{S}_{prev}$) and attempts to apply the three key operators to create new candidate assortments:

- **`add`:** For a given current assortment, select a restaurant not included in the set based on its profit contribution (i.e., the product of its revenue p_{qr}^d and attraction value v_{qr}^d). A weighted random selection favors higher-profit restaurants, prioritizing those with greater expected profitability.
- **`remove`:** For a given current assortment, eliminate a restaurant from the assortment based on its inverse profit contribution. Lower-profit restaurants are more likely to be removed.
- **`swap`:** Replace a restaurant in the current assortment with one not present in it, using the weighted selection rules from the `add` and `remove` operators.

To generate new assortments, we maintain a record of assortments that have already been included in the set of available assortments. This ensures that only genuinely new assortments are generated. If no further unique assortments can be created, the algorithm halts the generation process.

Algorithm 2 Generate New Assortments for Selected Assortments

Input: Data instance, previous optimal selected assortments $\mathcal{A}_{\text{prev}}$, current possible assortments $\mathcal{A}_{\text{poss}}$, set R_q^d.
Output: Updated possible assortments $\mathcal{A}'_{\text{poss}}$.
foreach $(d, q) \in \mathcal{A}_{\text{prev}}$ **do**
  Extract previously optimal selected assortment $\mathcal{S}_{\text{prev}}$
  Calculate scores for each restaurant in R_q^d based on p_{qr}^d and v_{qr}^d
  Attempt to `add`: Select a restaurant not in $\mathcal{S}_{\text{prev}}$ and add it `// Section: add`
  Attempt to `remove`: Remove a restaurant from $\mathcal{S}_{\text{prev}}$ `// Section: remove`
  Attempt to `swap`: Swap a restaurant in $\mathcal{S}_{\text{prev}}$ with one not in it `// Section: swap`
  Add new assortments to $\mathcal{A}'_{\text{poss}}$
return $\mathcal{A}'_{poss}$

Assortment removal.

The assortment removal pseudo-code in Algorithm 3 ensures that the set of available assortments remains efficient by removing outdated options. Each assortment tracks the last iteration in which it was used. If an assortment has not been utilized within a defined inactivity threshold φ, it is removed from the set of possible assortments. This approach streamlines the optimization process by focusing on active and relevant assortments, reducing computational overhead while maintaining the quality of solutions.

The IAG algorithm is a tailored heuristic approach designed to address the combinatorial complexity of the RSRDP. By iteratively generating and refining the set of possible assortments, the algorithm balances exploration and exploitation to efficiently search for near-optimal solutions. The three operators, `add`, `remove` and `swap`, play a central role in generating new assortments, leveraging profitability metrics to guide modifications

Algorithm 3 Remove Old Assortments Based on Inactivity

Input: Current possible assortments $\mathcal{A}_{\text{poss}}$, last used iteration $\mathcal{I}_{\text{last}}$, current iteration i, inactivity threshold φ.
Output: Updated possible assortments $\mathcal{A}'_{\text{poss}}$.

```
foreach (d, q) ∈ A_poss do
    foreach assortment S_q^d ∈ A_poss(q, d) do
        Extract last used iteration I_last(S_q^d) if i − I_last ≥ φ then
            Remove S_q^d from A_poss(q, d)
return A'_poss = A_poss
```

to the current solution. Additionally, an assortment removal mechanism ensures computational efficiency by eliminating outdated options based on inactivity, allowing the algorithm to focus on the most promising subsets. To further enhance its performance, we extend the IAG framework by introducing an operator scoring system inspired by the Adaptive Large Neighborhood Search (ALNS) algorithm. This extension integrates a scoring mechanism to evaluate and prioritize the operators, providing a structured way to adaptively guide the search process based on observed performance. In the next section, we present the tailored ALNS-inspired algorithm and explore its potential to improve upon the IAG's performance for larger and more complex instances.

5.3.2. Incorporating Adaptive Large Neighborhood Search scoring system

Building on the IAG framework, we seek to investigate whether the three assortment-generation operators (`add`, `remove`, and `swap`) differ in performance. To this end, we borrow the idea of a scoring mechanism from the Adaptive Large Neighborhood Search (ALNS) algorithm [Ropke and Pisinger, 2006]. ALNS is a metaheuristic commonly used for large-scale combinatorial optimization, where destroy-repair operators are dynamically weighted based on their observed success in improving the solution. By introducing a similar adaptive mechanism into the IAG algorithm, we can prioritize the operators that more frequently yield beneficial new assortments.

It is important to emphasize, however, that this extended version of the IAG is not a standard ALNS approach. In a classical ALNS algorithm, the solution is typically partially "destroyed" and then "repaired," with acceptance based on an improvement (or acceptance) criterion. The solution is not re-optimized in a master problem each time. Rather, neighborhoods are explored by modifying the solution directly and tracking improvements. In contrast, our IAG framework, whether extended with ALNS scoring or not, re-solves the restricted master problem in every iteration. Although the ALNS-inspired scoring scheme helps select which assortments to generate and keep, we still rely on full optimization to decide which assortments to use in each district-cuisine pair (d, q). Thus, the IAG remains a Benders-based iterative generation procedure at its core, where the ultimate decision regarding assortment selection is made by solving an optimization model, rather than purely by local acceptance rules.

Traditional ALNS relies on a cycle of destroying part of the solution and repairing it to explore different regions of the search space. In our tailored extension, the removal of inactive assortments assumes a light destructive role, while our generation operators (`add`, `remove`, and `swap`) act as repair operators. During each iteration, we randomly select one of these operators according to a weighted probability distribution and generate new assortments. If any of the newly added assortments improve the solution, we increase the score of the operator that produced them. This score update feeds back into the operator weights, causing more successful operators to be chosen more often in subsequent iterations.

Initially, all operator weights are set to 1, and operators are selected randomly. During each iteration, the selected operator's score is updated based on its contribution to improving the solution. Let Ω be the set of operators and for each operator $\omega \in \Omega$, and let w_ω represent its weight and σ_ω its score. Operator weights are then adjusted using the following formula:

$$w_\omega = (1 - \xi)w_\omega + \xi \frac{\sigma_\omega}{\sum_{\omega' \in \Omega} \sigma_{\omega'}} \tag{5.30}$$

Where $\xi \in (0, 1)$ is the reaction factor, controlling how sensitive the weights are to changes in the operators' performance. The updating rule for the scores is set as follows: σ_ω is increased by ζ if a new best solution is found. In each iteration, a repair operator is chosen to generate a new possible solution. To decide which operator is chosen at each iteration, we randomly pick one repair operator using probability:

$$p_\omega = \frac{w_\omega}{\sum_{w' \in \Omega} w_{\omega'}} \tag{5.31}$$

Algorithm 4 outlines the extended IAG approach with ALNS-inspired scoring. At each iteration, we (i) solve the current restricted problem using Benders decomposition, (ii) update the score of the chosen operator if it improved the solution, (iii) reweight the operators, and (iv) generate new assortments by the chosen operator. The removal of inactive assortments keeps the search space from growing uncontrollably. The process is iterated until either a fixed amount of CPU time T_{limit} is used or after a given number of iterations I_{limit}.

Algorithm 4 Extended IAG with ALNS Score-Based System

Input: Data instance, time limit T_{limit}, maximum iterations I_{limit}, reaction factor ξ, inactivity threshold φ, score ζ
Output: Best objective P^*, associated assortments $\mathcal{A}^*$, weights w_ω for operators
Initialize: Operators $\Omega = \{\texttt{add}, \texttt{remove}, \texttt{swap}\}$, weights $w_\omega \leftarrow 1\ \forall \omega \in \Omega$, scores $\sigma_\omega \leftarrow 0\ \forall \omega \in \Omega$
Set $P^* \leftarrow$ NBR objective, $\mathcal{A}^* \leftarrow$ NBR assortments, iteration counter $i \leftarrow 0$
while $i < I_{limit}$ ***and*** *runtime* $< T_{limit}$ **do**
  Solve the **Restricted Problem (RP)** using Benders decomposition
  Retrieve current objective P_i and selected assortments $\mathcal{A}_i$
  if $P_i > P^*$ **then**
    Update $P^* \leftarrow P_i$, $\mathcal{A}^* \leftarrow \mathcal{A}_i$
    Increase score of selected operator $\sigma_{\text{chosen\_op}} \leftarrow \sigma_{\text{chosen\_op}} + \zeta$
  Update operator weights: $w_\omega \leftarrow (1-\xi) w_\omega + \xi \frac{\sigma_\omega}{\sum_{\omega' \in \Omega} \sigma_{\omega'}}$
  Select next operator chosen_op $\in \Omega$ using probability p_ω
  Apply **operator** (`add`, `remove`, or `swap`) to generate new assortments
  Remove unused assortments based on inactivity threshold φ
  Increment iteration counter $i \leftarrow i + 1$
return P^*, $\mathcal{A}^*$, *and weights* $w_\omega\ \forall \omega \in \Omega$

This extended IAG framework with the score-based system enables dynamic evaluation of the operators, providing valuable insights into their effectiveness. By analyzing the final weights assigned to each operator, we can identify which operators contribute most to generating high-quality solutions. This information offers a deeper understanding of the solution space and helps refine the heuristic approach.

In conclusion, this chapter has introduced a range of resolution approaches to address the RSRDP, progressing from exact methods to heuristic algorithms. While exact methods provide a theoretical benchmark, their scalability limitations necessitate alternative approaches for larger problem instances. The proposed Iterative Assortment Generation (IAG) algorithm, and its extension with the ALNS-inspired score-based system (IAG-E), offer computationally efficient strategies that leverage problem-specific insights to achieve near-optimal solutions. In the next chapter, we move on to numerical experiments, comparing the performance of these algorithms and analyzing their effectiveness in solving both small-scale and large-scale problem instances. This evaluation provides critical insights into the practical applicability and strengths of each approach.

6
Results

In this chapter, we evaluate the performance of the proposed models and solution algorithms through an extensive set of computational experiments. The results are divided into two key sections: (1) computational performance, where we assess the efficiency of the proposed algorithms under varying instance settings, and (2) analyzing the impact of commission-based (CB) and fixed employment (FE) compensation policies on operational costs, rider utilization, and profitability under varying network configurations. Additionally, in Section 6.3, we test the proposed algorithm on Amsterdam restaurant data to find practical trade-offs between rider shift duration and delivery windows, comparing profitability, fleet size and rider workload measures.

The computational performance analysis examines three dimensions: (i) the effectiveness of algorithm components in reducing computational time, comparing standard Gurobi (`G`), Gurobi with Benders decomposition (`G+B`), and the Iterative Assortment Generation algorithm (`IAG`); (ii) the comparative performance of the `IAG` and its score-based extension `IAG-E` for smaller and medium-sized instances; and (iii) the profitability and efficiency of the integrated RSRDP with `IAG` versus the separated benchmark model explained in Section 5.1. For each category, we report metrics such as computation time, objective value, and improvement percentage across multiple scenarios.

Results show that the integrated RSRDP consistently outperforms the benchmark in expected profitability, while maintaining service quality standards, with `IAG` delivering scalable, high-quality solutions. The score-based extension `IAG-E` does not add additional value to the performance of the `IAG` algorithm under the experiments tested. Additionally, CB excels in high-variability environments, providing high profitability, and outperforming FE consistently on profitability measures. However, FE performs better in stable settings, ensuring balanced fleet utilization.

All computational experiments were conducted on two systems: smaller instances were solved on a virtual machine equipped with an Intel(R) Core(TM) i7-6700HQ CPU, 2.60 GHz processor, and 32 GB of RAM, while larger instances were processed on the DelftBlue supercomputer [Delft High Performance Computing Centre, 2024] with an Intel(R) Xeon(R) Gold 6248R CPU, 3.00 GHz processor, and 185 GB of RAM. All experiments were implemented in `Python 3.9.7` and solved using `Gurobi 11.0.1`.

6.1. Computational performance

6.1.1. Instance description

The algorithm was tested on samples of generated datasets using a set of instances. Each instance is defined by the tuple $(D, Q, M, R, v^d_{q0}, [y^L_{qd}, y^U_{qd}])$, where D is the number of service districts, Q the number of cuisine types, M the number of zones, and R the total number of restaurants available. The no-purchase option v^d_{q0} is considered in two configurations: either set to zero, meaning customers always place an order once they choose a cuisine, or assigned a value of 10, where customers have the possibility of opting out, better reflecting real-world behavior where some customers browse without committing to a purchase. The dissimilarity parameters $y^d_q \in [y^L_{qd}, y^U_{qd}])$ are sampled from the ranges $[0, 1]$, indicating competitiveness between within nest restaurants, and $[1, 2]$, indicating synergy between restaurants. Only for $\gamma^d_q = 1$ we have independence between restaurants corresponding to the MNL model. These configurations allow us to evaluate different behavioral scenarios and compare against the benchmark where it is assumed that $v^d_{q0} = 0$ and $\gamma^d_q \in [0, 1]$.

The experiments cover a broad range of problem sizes. Smaller instances include $Q = 2$ cuisine types and

restaurant counts $R \in \{10, 15, 20\}$, whereas larger instances explore $Q = 4$ with restaurant counts extending to $R \in \{50, 100, 150, 200\}$. The spatial structure remains fixed with $D = 4$ service districts and $M = 15$ zones, ensuring consistency across all test cases. Zones are generated using the H3 indexing system. Each instance is constructed to reflect real-world conditions, explained in the subsequent section.

Additionally, we evaluate the impact of set sizes on computational efficiency, we test three configurations: small, medium, and large, defined as follows: $T \in \{50, 100, 150\}$, $Q \in \{2, 4, 6\}$, $S \in \{2, 3, 4\}$, $R \in \{10, 20, 30\}$ and $M \in \{4, 15, 59\}$. For these experiments, we compare the performance of the exact method, the G+B approach, and the IAG algorithm. These tests use $\gamma_q^d \in [0, 1]$ and $v_{q0}^d = 0$. To ensure comparability, the smallest case in each configuration is treated as the baseline for evaluating the impact of scaling. Due to the computational complexity of the exact method, these experiments are limited to relatively small instances with fewer restaurants. By systematically varying the set sizes, we aim to identify trends in computational efficiency and the relative trade-offs between the different solving approaches.

Parameter settings.

We adopt realistic parameter settings derived from industry information to evaluate the algorithms' performance under practical scenarios. Table 6.1 summarizes these settings, including spatial and temporal parameters, demand distributions, and cost coefficients. The restaurant and customer distributions are configured to reflect realistic urban settings, with restaurants concentrated 50% in central zones and 50% randomly distributed across peripheral zones. For the parameters used in the score-based IAG extension (IAG−E), we follow Gansterer et al. [2021].

Table 6.1: Parameter settings for computational performance test instances.

Parameter	Value	Description
T	48 periods	Total time horizon
D	4 districts	Number of districts
M	15 zones	Number of spatial zones
ρ	$\max(\tau_{mm'})$ periods	Delivery window deadline
η	1 period	Meal preparation time
λ_{mt}	$\sim$ Poisson(2)	Customer demand distribution
$\theta_{\min}$	0 periods	Minimum rider shift duration
$\theta_{\max}$	48 periods	Maximum rider shift duration
$c_{FE}^{overhead}$	€54 euro	Daily overhead costs per rider under FE policy
$c_{CB}^{overhead}$	€18 euro	Daily overhead costs per rider under CB policy
c_{qr}^{d}	€3.75 euro	Delivery cost per order CB policy
c^{t}	€2.50 euro	Time discretized hourly wage for riders under FE policy
φ	5 iterations	Inactivity threshold value IAG algorithm
I_{limit}	50 iterations	Iteration limit IAG algorithm
T_{limit}	3600 seconds	Computational time limit IAG algorithm
ξ	0.4	reaction factor IAG−E algorithm
ζ	10	score update for new best solution IAG−E algorithm

Restaurant revenues p_{qr}^d and attraction levels v_{qr}^d are generated using the methodology of Alfandari et al. [2021]. We sample U_{dqr} from a uniform distribution over $[0, 1]$, and X_{qr}^d and Y_{qr}^d are independently sampled from a uniform distribution over $[5, 15]$. Then, the revenues and attraction levels are calculated as: $p_{qr}^d = 10 \times U_{qr}^d \times X_{qr}^d$, $v_{qr}^d = 10 \times (1 - U_{qr}^d) \times Y_{qr}^d \quad \forall d \in D,\ q \in Q,\ r \in R$, where higher-priced restaurants tend to have lower attraction levels, aligning with the idea that expensive options appeal to fewer customers. However, random variation ensures that this relationship is not strictly deterministic. The revenue distribution is skewed, producing many low-revenue restaurants and a few high-revenue ones, reflecting real-world restaurant dynamics. Platform revenue is modeled as a percentage (15–30%) of customer orders, consistent with industry standards. To generate realistic profits, customer order values are set based on an average of €34, with revenue and attraction parameters sampled accordingly. We do not impose any restrictions on minimum restaurants to be included in assortments.

6.1.2. Computational performance results

For all tests, we adopt the FE policy. A time limit of 3600 seconds is imposed on each instance. For instances that do not converge to optimality within this limit, we report the best-found solution and the corresponding optimality gap, as well as its iteration number for the IAG algorithm. We solve 20 samples of each instance, and present the average results. To evaluate the relative performance of different models, we calculate the average percentage improvement for key performance indicators (KPIs), such as computation time, objective value or required fleet size, using the following formula:

$$\text{KPI improvement}(\%) = \frac{1}{20}\sum_{I=1}^{20} 100\% \cdot \left(\frac{|\text{KPI}(1) - \text{KPI}(2)|}{\text{KPI}(1)}\right) \tag{6.1}$$

Here, $I = 1, ..., 20$ denotes each sample, with (1) representing KPI results from Method 1 that is compared to (2), representing results from Method 2.

The effect of different algorithm components.

Table 6.2 presents the objective values, computation times, and optimality gaps for all methods, along with the number of iterations for IAG. Table 6.3 quantifies the performance improvements, showing percentage reductions in time and any changes in objective values. The results highlight the impact of Benders decomposition and the IAG algorithm on solving the integrated model. While all three methods achieve the same objective values across tested instances, significant differences emerge in computation time. Adding Benders decomposition (G+B) substantially reduces computation time compared to using only the standard solver (G), with improvements of up to 94.87%. This effect is particularly evident in complex cases with more restaurants, where G struggles to close optimality gaps, despite reporting the correct solution. The decomposition effectively strengthens dual bounds, leading to faster convergence.

Further time reductions are observed for IAG, outperforming G+B by an additional 14% in scenarios with low outside utility values. However, when the outside utility was set higher, IAG requires more time than G+B, though all instances were still solved within three minutes. Notably, IAG scales well as the number of restaurants increases, benefiting from its ability to explore the solution space efficiently without full enumeration. The number of iterations remains low, averaging between 2 and 6, underscoring its rapid convergence. Interestingly, when complexity in terms of available restaurants increases, IAG performs relatively better, demonstrating that IAG scales well and provides a robust alternative to exact methods for mid-sized instances.

Table 6.2: The effect of Benders decomposition and IAG.

Instance	Gurobi			Gurobi + Benders			IAG			
	Obj. (€)	Time (s)	Gap (%)	Obj. (€)	Time (s)	Gap (%)	Obj. (€)	Time (s)	Gap (%)	#Iter.
(4,2,15,10,0,[0,1])	39267.59	139.65	0	39267.59	55.86	0	39267.59	47.37	0	2.5
(4,2,15,10,0,[1,2])	73512.25	1551.69	0	73512.25	80.38	0	73512.25	69.03	0	2.1
(4,2,15,10,10,[0,1])	30523.12	112.02	0	30523.12	54.86	0	30523.12	105.25	0	3.9
(4,2,15,10,10,[1,2])	50187.17	1329.27	1.32	50187.17	80.92	0	50187.17	151.46	0	3.9
(4,2,15,15,0,[0,1])	40434.99	1711.60	12.7	40434.99	106.09	0	40434.99	35.81	0	1.6
(4,2,15,15,0,[1,2])	75682.05	3293.44	900.72	75682.51	868.49	0	75682.51	25.15	0	1.5
(4,2,15,15,10,[0,1])	32154.20	1351.24	2.54	32154.20	67.75	0	32154.20	114.76	0	5.8
(4,2,15,15,10,[1,2])	53176.87	2778.13	332.23	53240.35	977.54	0	53240.35	194.16	0	4.5

Impact of set size on computational efficiency.

The heatmap of computational times presented in Figure 6.1 further highlights the scalability challenges faced by the exact methods as the problem size increases. For the G+B approach, the rapid increase in computational time with the number of restaurants (|R|) can be attributed to the growth in enumerations required to evaluate feasible solutions as the problem space expands. Additionally, the increase in computational time with the number of zones (|M|) reflects the added complexity of modeling rider movement and interactions across a larger spatial network. With more zones, the solver must account for an exponentially larger number of possible routing and delivery combinations, which drives up the computational burden. In contrast, the IAG algorithm exhibits more stable computational times, demonstrating its efficiency and suitability for larger-scale problems. This improvement stems from its intelligent selection of possible assortments, which effectively mitigates the computational challenges associated with restaurant complexity. However, as the number of zones

Table 6.3: Performance improvement summary of effects Benders decomposition and IAG algorithm.

Instance	Obj. % Impr. G → G+B	Obj. % Impr. G+B → IAG	Time % Impr. G → G+B	Time % Impr. G+B → IAG
(4,2,15,10,0,[0,1])	0%	0%	60%	15.20%
(4,2,15,10,0,[1,2])	0%	0%	94.82%	14.12%
(4,2,15,10,10,[0,1])	0%	0%	50.03%	-91.85%
(4,2,15,10,10,[1,2])	0%	0%	93.91%	-87.17%
(4,2,15,15,0,[0,1])	0%	0%	93.80%	66.25%
(4,2,15,15,0,[1,2])	0%	0%	73.63%	97.10%
(4,2,15,15,10,[0,1])	0%	0%	94.87%	-69.39%
(4,2,15,15,10,[1,2])	0.12%	0%	93.01%	80.14%

increases, the computational time also rises, likely due to the increased complexity of integrating the demand in the time-space graph. The expanded spatial structure necessitates tracking a greater number of potential rider movements and needs to integrate this with the different assortment decisions, thereby adding to the computational load despite the algorithm's more efficient handling of restaurant choices.

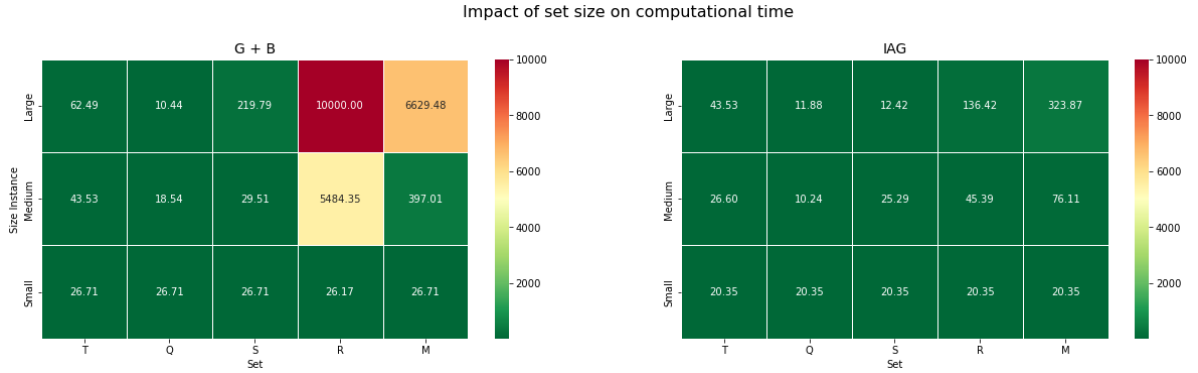

Figure 6.1: Comparison of different set sizes (small, medium, large) and their impact on computational time, for the exact G+B approach on the left, and the IAG approach on the right.

Comparing the integrated RSRDP to the separated benchmark.

The results in Table 6.4 compare the separated benchmark model, which solves assortment and rider optimization sequentially, with our integrated RSRDP solved with IAG, where both decisions are jointly optimized. The table presents the objective values, the number of riders required, and the percentage improvement in both metrics across different problem instances.

The integrated RSRDP model consistently outperforms the separated benchmark in terms of expected profit, with improvements ranging from marginal increases of 0.09% in simpler cases to substantial gains exceeding 28% in more complex settings. The largest improvements occur when the outside utility is high ($v^d_{q0} = 10$) and the restaurants exhibit higher synergy among each other ($\gamma^d_q \in [1, 2]$). This indicates that the benefits of integration grow as users have an option of not purchasing anything when they have already chosen a cuisine type or restaurants are synergistic with respect to each other, or both. Conversely, in cases with no outside utility and high competitiveness, the profit improvements remain modest, often below 1%, the value of joint optimization is less pronounced yet still beneficial. For the, in our opinion, most real-world reflective case of positive outside-utility ($v^d_{q0} = 10$) and competitiveness between restaurants ($\gamma^d_q \in [0, 1]$), the improvements are still significant around 6%. Overall, the impact of integration becomes stronger as the problem size increases, indicating that for a larger real-world network, integration is even more beneficial.

In terms of fleet size, the integrated RSRDP generally reduces the number of required riders, with decreases of up to 13.86%. This efficiency gain results from the coordinated optimization of assortment and delivery, leading to a more compact and effective allocation strategy. However, in some instances, particularly those with high outside utility, the required fleet size increases, sometimes even by over 50%. This is driven by the RSRDP's ability to attract more customers through strategically optimized assortments, leading to a higher order volume

Table 6.4: Comparing the separated benchmark to the integrated model.

Instance	Separated benchmark		Integrated model		% Impr.	
	Obj. value	# riders	Obj. value	# riders	Obj. value	# riders
(4,2,15,10,0,[0,1])	38993.05	27.34	39267.49	23.55	0.70	-13.86
(4,2,15,10,0,[1,2])	72768.28	42.96	73512.25	37.54	1.01	-12.62
(4,2,15,10,10,[0,1])	29612.76	21.96	30523.12	21.71	2.98	-1.14
(4,2,15,10,10,[1,2])	43045.87	25.78	50187.17	28.63	14.23	11.06
(4,2,15,15,0,[0,1])	40399.01	24.87	40435.00	24.57	0.09	-1.21
(4,2,15,15,0,[1,2])	75423.98	43.47	75682.51	40.33	0.34	-7.22
(4,2,15,15,10,[0,1])	31491.59	19.94	32154.20	21.28	2.06	6.72
(4,2,15,15,10,[1,2])	49643.69	26.83	53240.35	33.41	6.76	24.52
(4,2,15,20,0,[0,1])	48635.53	24.35	48730.63	23.91	0.20	-1.81
(4,2,15,20,0,[1,2])	87078.07	40.98	87992.27	39.99	1.04	-2.42
(4,2,15,20,10,[0,1])	37240.32	19.00	39235.44	23.72	5.08	24.84
(4,2,15,20,10,[1,2])	52769.58	25.79	62535.25	34.73	15.6	34.66
(4,4,15,50,0,[0,1])	67906.86	29.39	67980.02	28.73	0.11	-2.25
(4,4,15,50,0,[1,2])	104933.61	41.46	105252.31	39.31	0.30	-5.19
(4,4,15,50,10,[0,1])	48353.21	21.45	52016.95	25.77	7.04	20.14
(4,4,15,50,10,[1,2])	57285.94	24.83	69201.41	30.64	17.22	23.40
(4,4,15,100,0,[0,1])	78849.35	31.34	79029.39	29.90	0.23	-4.59
(4,4,15,100,0,[1,2])	118097.91	41.00	118216.01	38.32	0.10	-6.54
(4,4,15,100,10,[0,1])	59575.97	24.08	64966.45	28.57	8.30	18.65
(4,4,15,100,10,[1,2])	62437.89	20.85	82351.77	31.55	24.18	50.32
(4,4,15,150,0,[0,1])	82233.79	29.47	82302.65	28.69	0.08	-2.65
(4,4,15,150,0,[1,2])	123268.25	39.28	123296.26	38.36	0.02	-2.34
(4,4,15,150,10,[0,1])	62038.43	22.71	67892.55	27.93	8.62	22.99
(4,4,15,150,10,[1,2])	61038.09	20.24	85513.95	31.51	28.62	55.68
(4,4,15,200,0,[0,1])	80393.57	31.15	80525.82	28.99	0.17	-6.93
(4,4,15,200,0,[1,2])	123371.40	43.27	123617.42	40.41	0.20	-6.61
(4,4,15,200,10,[0,1])	62801.29	24.73	68775.88	28.67	8.69	15.93
(4,4,15,200,10,[1,2])	69387.55	21.56	85378.27	33.84	18.73	56.96

that necessitates additional riders. Despite this, the corresponding profit increase is substantial, with instances such as (4,4,15,200,10,[1,2]) showing an 18.73% improvement in profit alongside a 56.96% rise in fleet size. These cases highlight that while more riders are required, the increase is a direct consequence of capturing more market demand and driving higher overall profitability.

Overall, the performance of the integrated approach is strongly influenced by the behavioral characteristics of customers. When customers are more likely to opt out of purchasing, integrating assortment and allocation decisions allows the platform to strategically influence demand, leading to higher revenues and sometimes requiring a larger fleet to meet demand. Similarly, when restaurants are synergistic, customers display stronger preferences for specific options, making optimized assortments significantly more valuable. On the other hand, when outside utility is low and restaurants display higher competitiveness, the impact of integration is limited, yet still beneficial in terms of profitability and reduced fleet size.

Impact of score-based extension to IAG algorithm.

In the analysis of the `IAG-E`, the operator weight distributions reveal some interesting patterns. As shown in Table 6.5, the operators are selected with relatively similar frequencies across scenarios, indicating that the `IAG-E` does not exhibit a strong preference for any one operator. However, there are some differences depending on the scenario. For example, the `remove` operator often has a slightly higher weight in simpler scenarios, e.g. $v^d_{q0} = 0,\ \gamma^d_q \in [0,1]$, suggesting that removing elements from the current solution is more effective in these cases. This makes sense since under the condition of $\gamma^d_q \in [0,1]$, where restaurants are competing more with each other under the nested logit framework. On the other hand, for scenarios with higher outside utility ($v^d_{q0} = 10$), the `add` and `swap` operators dominate, reflecting the increased importance of exploring alternative configurations in more complex settings, consistent with the synergistic nature of having $\gamma^d_q \in [1,2]$.

Although the IAG-E approach provides flexibility and balance in operator selection, it also leads to increased computational times, as indicated by the percentage increases reported in Table 6.5. The computation time increase can likely be attributed to the additional iterations required in this extension. Unlike the IAG implementation, where all three operators could be applied within a single iteration, the IAG-E applies only one operator per iteration and evaluates its performance before proceeding. This more granular evaluation strategy leads to additional iterations before convergence, thus increasing computational effort. Overall, the relatively even distribution of operator weights suggests that our IAG algorithm is flexible and well-balanced, as the IAG-E indicates that no single operator is universally superior across all scenarios.

Table 6.5: Optimized weight percentages of operators in IAG-E.

Instance	% Time incr.	w_a	w_r	w_s
(4,2,15,10,0,[0,1])	55.75	27.64	**36.91**	35.45
(4,2,15,10,0,[1,2])	43.82	32.13	33.03	**34.83**
(4,2,15,10,10,[0,1])	46.95	**33.49**	29.47	33.04
(4,2,15,10,10,[1,2])	69.79	**39.97**	27.20	32.83
(4,2,15,15,0,[0,1])	44.06	33.47	**35.96**	30.56
(4,2,15,15,0,[1,2])	85.63	30.49	34.35	**35.16**
(4,2,15,15,10,[0,1])	53.91	**42.84**	23.79	33.37
(4,2,15,15,10,[1,2])	62.75	35.43	26.02	**38.54**
(4,2,15,20,0,[0,1])	60.51	27.13	32.91	**39.95**
(4,2,15,20,0,[1,2])	70.03	28.3	**40.03**	31.67
(4,2,15,20,10,[0,1])	78.23	38.35	17.96	**43.70**
(4,2,15,20,10,[1,2])	57.50	**52.21**	19.44	28.35

In conclusion, the proposed IAG algorithm demonstrates a clear advantage in solving integrated optimization problems for meal delivery platforms. By achieving rapid convergence, maintaining scalability, and consistently delivering better results compared to separated models, the IAG algorithm provides a robust framework for enhancing profitability and operational efficiency in the dynamic and competitive meal delivery industry.

6.2. Impact of compensation policies and other managerial insights

6.2.1. Experimental setup

We evaluate commission-based (CB) and fixed employment (FE) compensation policies over a simulated 12-hour operating window (11:00–23:00) in a mid-sized urban area. The region consists of 34 zones across four service districts, with 100 restaurants spanning 10 cuisine types. Restaurant distribution reflects real-world conditions, with 50% in central zones and 50% in peripheral areas.

Customer arrivals follow a Poisson process with demand peaking during meal times, resulting in 1,250–1,500 daily orders. Assortments assume competitive restaurant interactions ($\gamma_q^d \in [0, 1]$), and customers may opt out at the cuisine level ($v_{q0}^d = 10$). Meal preparation takes 10 minutes, and deliveries must be completed within one hour. Rider shifts range from 4 to 12 hours. Under FE, riders earn €15 per hour (€2.50 per 10-minute time step), while under CB, they receive €5 per completed delivery, assuming an average of three deliveries per hour. Hiring costs range from 10–30% of wages, totaling €54 for platform-employed and €18 for independent riders. The integrated RSRDP is solved using `IAG` with a 2-hour time limit and a maximum of 50 iterations. Each scenario is replicated 10 times to ensure robust results.

6.2.2. Influence of network configurations on performance of CB and FE policies

We investigate the impact of several factors on the relative performance of CB and FE, including customer arrival rate fluctuations, restaurant distribution, shift regulations, and delivery deadlines. Summaries of the results can be found in the Tables presented in Appendix B.

Impact of restaurant distribution

We investigate how varying spatial distributions of restaurants influence platform performance under the FE and CB policies. Three scenarios are examined: the base configuration (50% central clustering), the centered scenario (90% concentrated within five central zones), and the distributed scenario (even dispersion across zones), as visualized in Figure 6.2. The results, presented in Figure 6.3, clearly demonstrate CB's consistent financial superiority across all restaurant distributions, driven primarily by its proactive rider management and adaptive scalability.

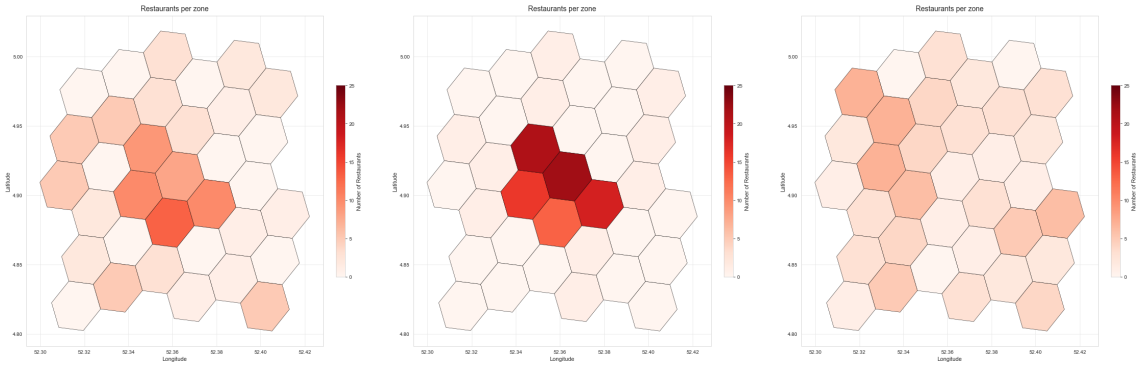

Figure 6.2: Heatmaps representing examples of restaurant distribution over operating area, base case (left), centered (middle) and distributed (right).

CB achieves higher revenues and profits through strategic rider relocation, allowing the platform to effectively anticipate and respond to spatial shifts in demand, as restaurants become more evenly distributed along the cases. Consequently, the platform opens more restaurants, increasing order volumes and enhancing customer satisfaction by offering broader choice. While CB's proactive relocation inevitably leads to increased delivery activities and associated operational costs, the policy effectively mitigates these expenses through lower per-rider hiring costs and the absence of direct relocation fees. This dynamic scalability ensures that CB's operational cost structure remains competitive, even as restaurant distributions become increasingly dispersed and operational complexity grows.

In contrast, FE faces significant trade-offs due to its salaried rider framework. Although FE achieves lower direct delivery and relocation expenses by restricting rider movement, the requirement to maintain sufficient salaried staff substantially increases fixed hiring costs, particularly as restaurant distribution becomes more geographically dispersed. The lack of dynamic repositioning severely limits FE's ability to capture emerging demand

opportunities effectively, resulting in consistently lower revenues and profitability compared to CB.

Overall, CB consistently outperforms FE in expected profit and revenue by flexibly scaling rider participation and repositioning at low marginal cost. In dispersed areas, CB mobilizes more riders on demand, boosting gains. However, with centralized or uniform restaurant layouts, FE's stable pool of salaried riders can maintain coverage without excessive idle costs, though it lacks CB's dynamic responsiveness.

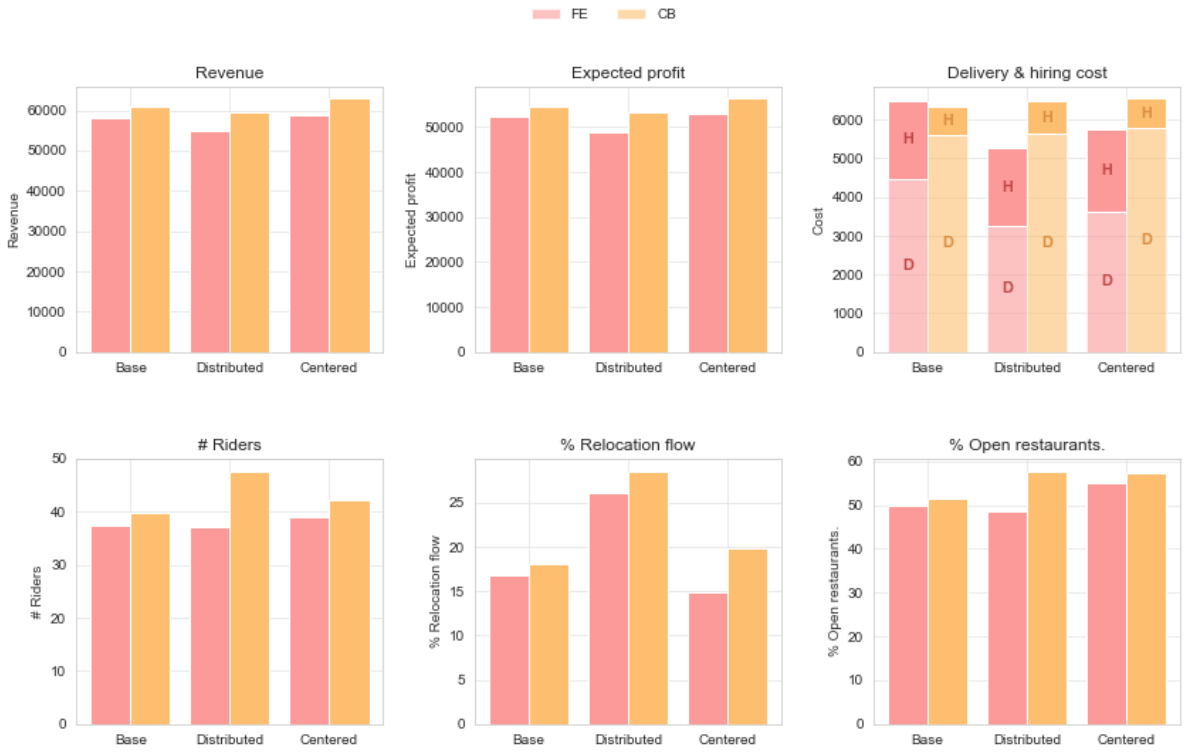

Figure 6.3: Plots of performance of the FE and CB policy on different metrics for evaluating the impact of restaurant distributions.

Impact of shift durations

To investigate how varying rider shift durations shape platform performance, seven distinct shift windows are evaluated, where each shift window represents the average minimum and maximum shift durations $[\theta_{min}, \theta_{max}]$ ranging from a fully flexible [0,12] schedule to a strict [6,6] window. Figure 6.4 presents the performance on the metrics for both business policies.

When riders have maximum flexibility ([0,12]), CB significantly outperforms FE, generating higher revenues and profits by leveraging dynamic relocation and flexible rider deployment. riders under CB are incentivized to reposition frequently, increasing order fulfillment and restaurant openings, which boosts revenues despite elevated delivery-related expenses. CB effectively controls overall costs by maintaining lower per-rider hiring expenditures, which offset the increased operational complexity stemming from frequent relocations. In contrast, FE's salaried rider structure ensures predictable costs but limits responsiveness, yielding lower revenues and fewer relocations, thus constraining profitability.

Interestingly, as shift windows become moderately flexible (e.g., [5,7]), profitability peaks for both policies. Moderate flexibility strikes an optimal balance: riders are available for sufficient time periods to manage transient demand spikes without incurring excessive idle times or inflated hiring costs. FE achieves cost containment through predictable staffing levels, while CB continues to exploit moderate flexibility through targeted repositioning, optimizing rider productivity and revenues.

Under strict shift durations ([6,6]), both policies rely more heavily on strategic relocations to leverage limited rider availability effectively. CB continues to generate higher revenues and profits despite substantial relocation activities and associated delivery costs, whereas FE demonstrates a clearer cost advantage by maintaining a smaller, predictably staffed workforce.

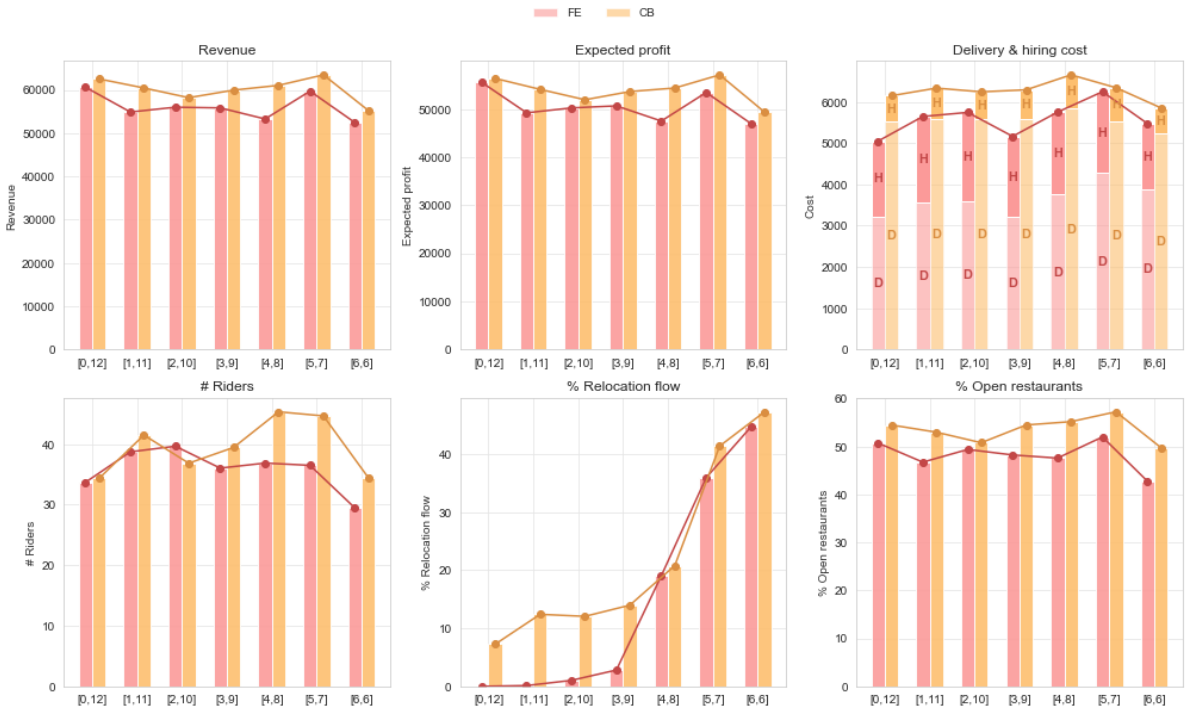

Figure 6.4: Plots of performance of the FE and CB policy on different metrics for evaluating the impact of shift durations.

Across these scenarios, the percentage of open restaurants also varies, reflecting how the platform balances wider restaurant availability against the delivery and hiring costs that accompany more dispersed operations. Generally, CB's flexibility encourages opening a higher fraction of restaurants, betting that enough riders will shift to meet new pockets of demand. FE, meanwhile, more carefully weighs the benefit of opening additional restaurants against the need to staff them with salaried riders.

Ultimately, flexible shift policies significantly enhance platform performance through increased responsiveness, particularly under CB. However, moderate shift rigidity ([5,7]) can yield optimal profitability for both FE and CB, balancing operational agility with predictable cost management. Strictly rigid shifts provide predictability but limit responsiveness, favoring platforms prioritizing cost stability over revenue maximization.

Impact of delivery time windows

Maximum delivery deadlines are an important measure for customer satisfaction, often being the counterpart of cost minimization. We investigate the impact of different maximum delivery deadlines, or delivery windows, and the results are presented in Figure 6.5. Short delivery deadlines (20–30 mins) drastically constrain the operational radius, limiting restaurant openings and reducing order fulfillment. Consequently, revenues and expected profits drop significantly for both models. FE experiences pronounced financial strain at tight deadlines, given its obligation to maintain a sufficient salaried workforce to quickly serve nearby customers, resulting in disproportionately high fixed labor costs.

As delivery deadlines extend (40–80 minutes), both FE and CB substantially increase geographic coverage and expand their restaurant assortments, significantly boosting order volumes and revenues. CB is particularly adept at exploiting these longer deadlines through dynamic rider repositioning and flexible workforce scaling. Despite higher relocation expenses and increased rider deployments, CB's lower hiring costs and agile workforce management translate into consistently higher profitability and revenues. FE also benefits from extended deadlines but experiences reduced gains due to fixed hiring expenses, which limit flexibility and scalability.

Interestingly, both policies demonstrate an initial increase in the number of riders when extending deadlines to about 40–50 minutes, after which the rider count stabilizes or gradually decreases. This reflects diminishing marginal returns, as the platform eventually reaches optimal geographic coverage, reducing the need for further rider scaling.

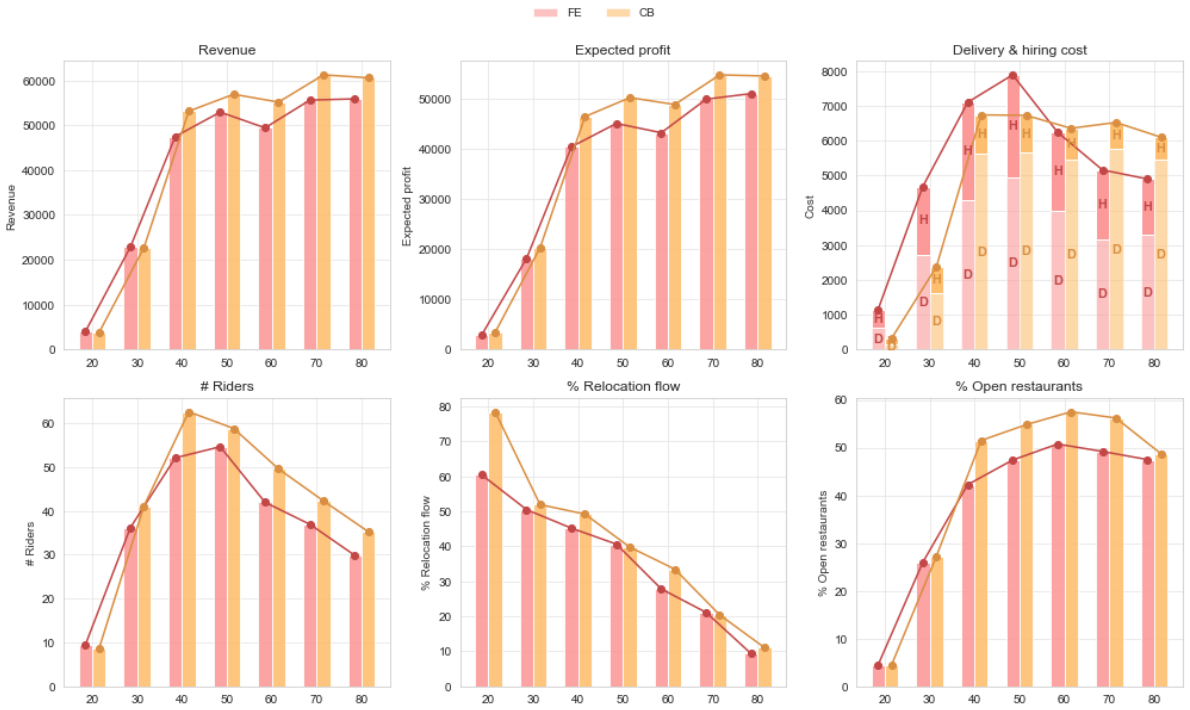

Figure 6.5: Plots of performance of the FE and CB policy on different metrics for evaluating the impact of maximum delivery deadlines.

While deadlines beyond 80 minutes can slightly enhance profitability, extended delivery times risk compromising customer satisfaction and eroding the platform's value proposition of timely service. Therefore, optimal delivery deadlines typically range between 40–70 minutes, balancing profitability, cost efficiency, and customer expectations. CB clearly demonstrates greater flexibility and profitability within this range, strategically aligning rider deployments with fluctuating demand. FE remains a viable alternative for platforms prioritizing predictable labor costs but consistently underperforms in responsiveness and overall profitability compared to CB, but requires a smaller fleet size.

Impact of customer arrivals throughout the time horizon

We next examine the influence of three different customer arrival distributions on meal delivery platform outcomes: base, uniform, and peak. In the base case, demand rises moderately around standard meal times without dramatic surges. By contrast, uniform arrivals spread demand consistently across the day, minimizing high demand intervals. The peak scenario features pronounced spikes at lunch and dinner. Figure 6.6 offers a visual representation of the key metrics across the three scenarios.

Under the base scenario, CB leverages dynamic rider repositioning to capitalize effectively on moderate demand surges, significantly increasing revenues and profits relative to FE. CB incurs higher costs due to proactive repositioning but effectively manages total expenses by maintaining lower rider hiring costs. FE's salaried workforce structure limits responsiveness, resulting in stable but lower overall revenue due to missed demand-capture opportunities during moderate peaks.

When demand is uniformly distributed throughout the operational horizon, the value of CB's dynamic repositioning diminishes, as fewer peak demand opportunities exist to exploit. Both FE and CB require minimal relocations, stabilizing operational costs. However, CB continues to marginally outperform FE, optimizing rider deployment more efficiently, whereas FE benefits from predictable labor expenditures and reduced idle times, narrowing the profitability gap. Under uniform demand, both policies achieve relatively stable profitability, with FE gaining slight competitiveness due to cost predictability.

In contrast, the peak scenario, with significant lunch and dinner surges, clearly highlights CB's strengths. CB dynamically mobilizes riders precisely during these high-demand windows, capturing substantial revenues despite escalating hiring and relocation costs. FE, restricted by fixed staffing levels, must significantly scale its salaried workforce to meet peak intervals, causing sharp increases in total labor expenses and limiting overall profitabil-

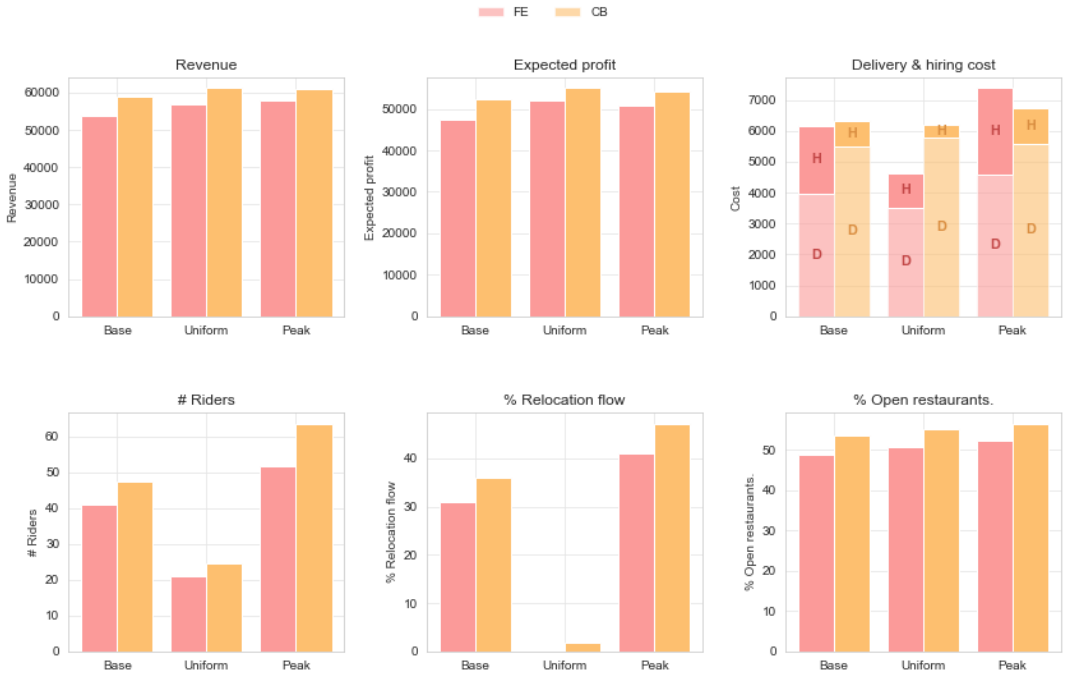

Figure 6.6: Plots of performance of the FE and CB policy on different metrics for evaluating the impact of various customer arrival distributions.

ity compared to CB. FE utilizes less riders but cannot match CB's agility and revenue capture during concentrated demand peaks.

Thus, CB consistently excels in volatile demand scenarios, maximizing profitability through flexible rider repositioning. FE performs best under steady demand conditions, prioritizing predictable cost structures but at the expense of responsiveness and higher revenue opportunities during peak periods.

Impact of business policy costs

This section explores how raising the delivery cost c^d_{qr} in the CB policy influences overall platform outcomes compared to FE, whose cost structure remains unchanged. Figure 6.7 summarizes key metrics for values of c^d_{qr} ranging from €3 through €7. Results clearly demonstrate CB's robust advantage across nearly all evaluated delivery-cost scenarios. Even as CB's variable per-delivery expenses significantly rise, the policy consistently maintains higher profitability and revenues compared to FE by strategically adapting rider deployments and selectively adjusting restaurant availability.

At lower delivery costs (e.g., €3-€3.5), CB effectively leverages flexible rider scheduling and repositioning to maximize profitability without substantially increasing delivery-related expenses. As per-delivery fees rise toward €7, CB dynamically fine-tunes its rider workforce and selectively adjusts open restaurant numbers, carefully balancing revenue opportunities against higher operational expenditures. Although this strategic flexibility results in higher total delivery costs, the marginal revenue gains consistently surpass the incremental expenses.

Relocation flows under CB display variability as delivery costs increase, reflecting strategic adjustments in rider deployment driven by the cost-benefit considerations of relocations versus additional rider hiring. Despite this variability, CB consistently maintains profitability advantages over FE due to its superior adaptability. At extremely high per-delivery costs, CB's advantage diminishes slightly, suggesting that in such scenarios, a hybrid model incorporating both salaried and flexible riders might optimize cost-efficiency and responsiveness.

Conclusion

Overall, our analysis reveals that under the conducted experiments, the CB policy generally outperforms the FE contracts in terms of profitability, particularly in environments characterized by high demand variability, broad service areas, or extended delivery windows. Under CB, the per-delivery compensation structure provides strong incentives for riders to relocate frequently and serve dispersed restaurants, thereby capturing more orders and

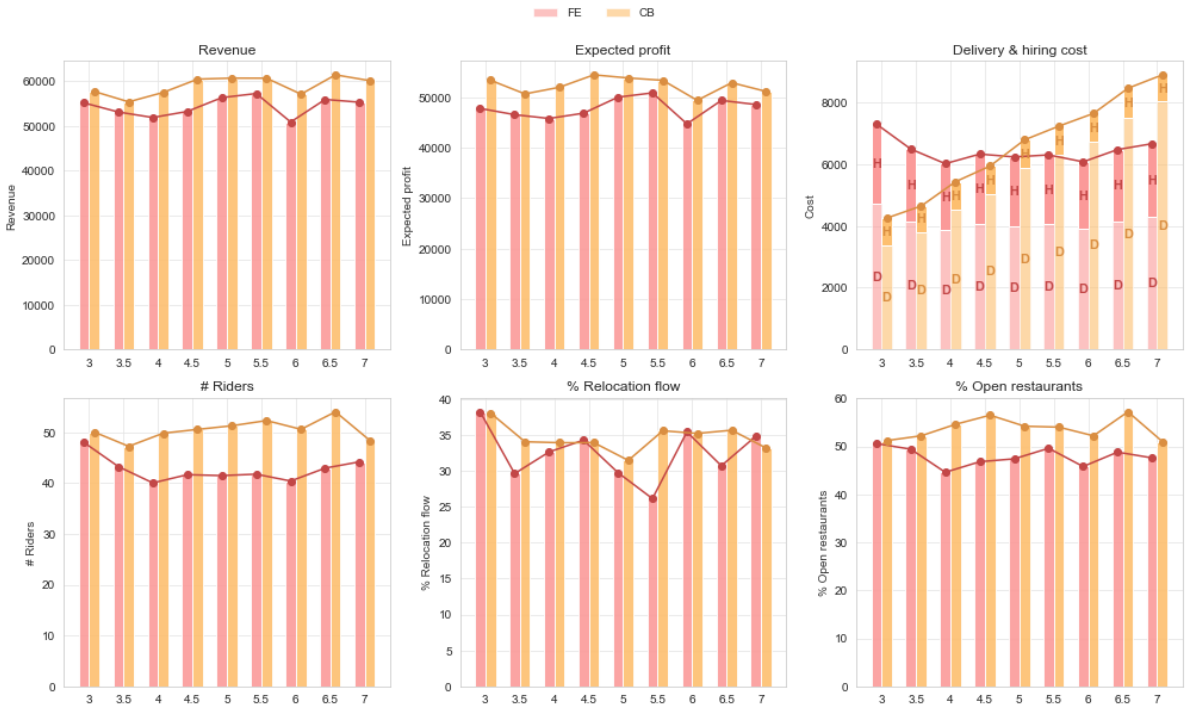

Figure 6.7: Plots of performance of the FE and CB policy on different metrics for evaluating the impact of changing c^d_{qr}.

boosting platform revenues. By contrast, FE provides more predictable costs and staffing coverage but can become expensive in volatile settings, since salaried riders must be paid regardless of demand fluctuations.

When considering the full range of parameters (e.g., rider shift lengths, restaurant distributions, wage overhead, and demand patterns), certain parameter extremes reinforce each other and further highlight the advantages of CB. For instance, very short shifts, widely scattered restaurants, high rider wages, and spiky (highly variable) demand can together amplify CB's relative profitability. Meanwhile, more moderate or uniform parameter values yield narrower differences. In other words, synergy, or friction, arises when multiple factors push in the same direction, favoring one policy over the other.

From a managerial perspective, these insights underscore that there is no one-size-fits-all solution. CB tends to excel in maximizing revenue but requires careful management of high-frequency rider relocations, which can elevate rider dissatisfaction. FE offers more predictable labor costs and steadier coverage, especially in uniform-demand or centralized-restaurant scenarios, yet its fixed cost structure may rapidly inflate under high demand. In practice, hybrid models that blend the dynamic scalability of CB with the cost stability of FE may offer a promising balance, potentially improving profitability, enhancing rider retention, and maintaining customer satisfaction across diverse operational settings.

6.3. Scaling RSRDP to real-world data

In this section, we evaluate the practical application of the RSRDP by scaling our proposed approach to real-world data from Amsterdam, The Netherlands. Using data from 100 restaurants, randomly selected to represent a diverse sample spanning 13 cuisines, we analyze how our model performs under realistic urban conditions. The data is obtained from a real meal delivery platform available in Amsterdam and the data includes specifics on the location of the restaurant, main cuisine type, review score (1.0-5.0) and price category (1-3). The study area is defined by the geographical boundaries of Amsterdam, which we partition into five service districts based on existing municipal divisions [Amsterdam, 2024b]. Each district is covered with the hexagonal zonal structure generated by the `H3` spatial indexing system at a resolution that approximates an 8-minute travel time between adjacent zone centroids.

To simulate customer demand, we combine population density factors [Amsterdam, 2024a] with temporal patterns that mirror typical meal-ordering behavior. Specifically, the arrival rate in each zone and time period, λ_{mt},

is computed based on these factors, resulting in an aggregate of roughly 2600 orders over the planning horizon. The attractiveness of restaurants is modeled through a regression that accounts for both price category and review scores. Here, the average price for a restaurant is determined by its price category, with:

$$\text{avg\_price}_r = \begin{cases} 10 & \text{if price category } 1 \\ 15 & \text{if price category } 2 \\ 30 & \text{if price category } 3 \end{cases} \tag{6.2}$$

The restaurant-specific attraction value is based on the negative influence of higher prices, and positively influenced by high review scores, given by:

$$v_{qr}^d = v_{q0}^d + 10 \times (2 \times \text{review}_r - \text{price\_category}_r + \epsilon_{dqr}^{(1)}) \tag{6.3}$$

Where review_r denotes the review score of restaurant r, and $\epsilon_{dqr}^{(2)} \sim$ extreme value type I $=$ Gumbel$(\mu = 0, \beta = 1)$ accounts for unobserved factors. Revenues from restaurants are based on the price category, using the commission rate of 15%-30% per order. We draw commission rates for restaurant r from the uniform distribution: $\text{commission\_rate}_r \sim U(0.15, 0.30)$. The price parameter is then calculated as:

$$p_{qr}^d = 2 \times \text{avg\_price}_r \times \text{commission\_rate}_r \times (1 + \epsilon_{qrd}^{(2)}) \tag{6.4}$$

Where the multiplicative noise term is $\epsilon_{qrd}^{(2)} \sim N(0, 0.05)$. This formulation means that the base price is scaled by 2 and then adjusted for the commission and a small normally distributed perturbation. Other parameters and sets are consistent with those described in Section 6.2, and Figure 6.8 provides an overview of the service districts, zonal structure, demand distribution, and restaurant locations, for the Amsterdam case study.

The selected case study reflects the typical urban distribution of restaurants, with a high concentration in the city center. Through this analysis, we investigate the impact of varying average shift durations and maximum delivery windows on key performance metrics, including platform profitability, rider workload per hour, and the required fleet size. Our objective is to identify a Pareto-optimal trade-off that can inform decision-making for meal delivery platforms. The RSRDP is solved under the Fixed Employment (FE) policy to evaluate its implications in this setting.

We define each solution as a vector $(Profit, \#R, WL)$, where we want to maximize the $Profit$, minimize the number of required riders $\#R$, and either minimize or have a reasonable workload WL for the riders, presented as the average number of orders per hour per rider. Mathematically, we say that a solution $(Profit_i, \#R_i, WL_i)$ dominates another solution $(Profit_j, \#R_j, WL_j)$ if the following conditions hold:

$$Profit_i \geq Profit_j \qquad \#R_i \leq \#R_j \qquad WL_i \leq WL_j \tag{6.5}$$

With at least one of these equalities being strict. In other words, $(Profit_i, \#R_i, WL_i)$ is considered better than $(Profit_j, \#R_j, WL_j)$ in all objectives, without being worse in any. The Pareto frontier consists of all solutions that are not dominated by any other solution in the set.

Table 6.6: Case study results.

		Delivery window [minutes]														
		30			40			50			60			70		
		Profit	# R	WL	Profit	# R	WL	Profit	# R	WL	Profit	# R	WL	Profit	# R	WL
Average	[4-6]	13263.07	87.23	3.64	15471.96	96.30	3.55	20274.53	76.79	4.66	23443.93	57.96	6.95	24045.89	57.36	7.20
shift	[5-7]	13140.89	86.52	3.14	14884.17	91.98	3.10	19177.51	76.68	3.89	22446.59	59.42	5.28	22761.63	46.85	6.37
duration	[6-8]	11801.41	85.85	2.70	16265.18	62.13	3.85	18044.48	68.08	3.69	21634.53	53.53	4.91	23029.38	47.96	5.69
[hours]	[7-9]	10691.79	64.68	2.65	13371.34	53.14	3.43	18805.00	64.36	3.56	21421.72	56.59	4.26	22315.70	46.85	4.77
	[8-10]	9583.14	64.16	2.34	12956.40	72.42	2.57	16942.20	70.62	2.90	20851.20	53.79	3.99	21659.90	44.83	4.43

The experimental evaluation examines the effects of varying two key operational parameters: the average rider shift duration and the maximum delivery window. Table 6.6 summarizes the outcomes in terms of platform profitability, the required number of riders (denoted as #R), and the average hourly workload, i.e. number of deliveries, per rider (WL). For instance, with a shift duration of 4–6 hours, increasing the delivery window from 30 to 70 minutes results in a profit increase from $13,263.07 to $24,045.89, yet the rider workload also rises from 3.64 to

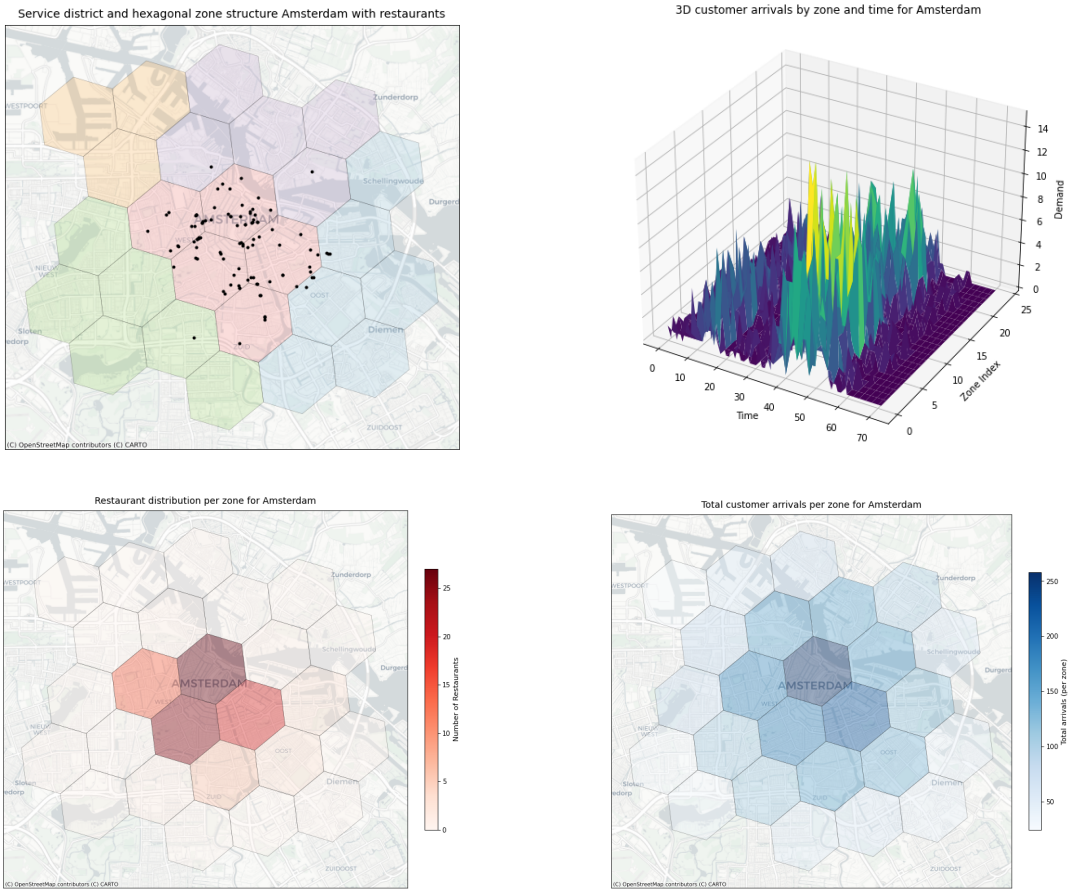

Figure 6.8: Case study data of Amsterdam. Top left indicates service regions, zones and restaurant locations. Top right presents customer arrivals over space and time for the time horizon. Bottom left shows heatmap of restaurant locations, bottom right heatmap of total customer arrivals in space over time horizon.

7.20 deliveries per hour. In contrast, longer shift durations tend to yield lower overall profitability but generally correspond to lower workloads, highlighting a trade-off between rider efficiency and cost-effectiveness.

The analysis is further enriched by the identification of Pareto frontier solutions. These solutions reveal balanced configurations where incremental increases in profitability are accompanied by relative changes in fleet size and rider workload. Figure 6.9 presents contrasting facets of the Pareto frontier solutions in a different context. The left subplot showcases the Pareto frontier considering profit, number of riders, and workload, where each point is color-coded to reflect workload intensity. For example, one Pareto optimal solution with a 4–6 hour shift and a 60-minute delivery window yields a profit of $23,443.93 with 57.96 riders and an average workload of 6.95 deliveries per hour, while another solution with a 5–7 hour shift and a 50-minute window achieves $19,177.51 profit with 76.68 riders and a workload of 3.89 deliveries per hour. Such comparisons underscore that a moderate delivery window, paired with an appropriate shift duration, can enhance service quality by keeping rider workloads within a reasonable threshold (around 3 deliveries per hour) while simultaneously improving profitability. The right subplot narrows the focus to the Pareto optimal solutions based solely on profit and the number of riders, while still employing workload for visual clarity. This simplification underscores the pivotal role of workload management in balancing service quality and operational efficiency. Notably, solutions situated in the bottom-right quadrant of both subplots exemplify the platform's capacity to maximize profitability with fewer riders, albeit with varying workload implications and a large delivery window.

These findings provide actionable insights for meal delivery platforms. In practice, shorter delivery windows enhance customer satisfaction and service quality, but may necessitate either more riders and lower profits. Conversely, extending the delivery window improves profit margins but risks overburdening riders and compromising timely service. For decision makers, the Pareto frontier serves as a decision-support tool, enabling a trade-off analysis where relative changes can be carefully evaluated.

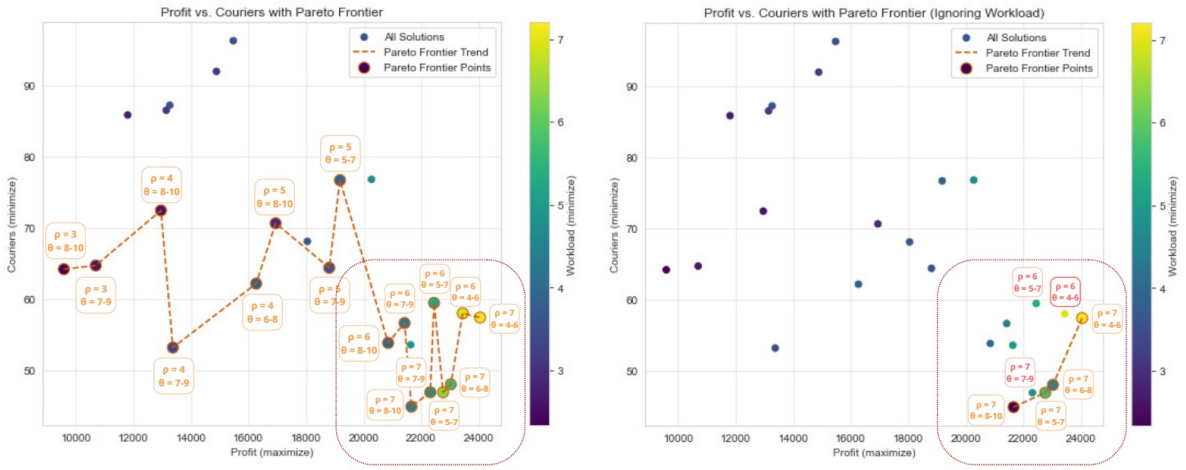

Figure 6.9: The 2D Pareto frontier solutions display profit on the x-axis and the number of riders on the y-axis, with color representing workload. The left figure considers dominance across all three metrics, profit, number of riders, and workload, while the right figure accounts only for profit and number of riders.

7

Conclusion and recommendations

Over the course of this thesis, we addressed the central research question: *How can we design the services for a meal delivery platform to maximize expected profits and solve it for large-scale systems?* To investigate this question, three research objectives were set: developing a mathematical model to capture the core decisions and constraints of the problem, designing and implementing efficient solution methodologies for both small and large instances, and analyzing the managerial implications of different compensation policies and network configurations on platform profitability and rider dimensioning.

By formulating the Restaurant Selection and Rider Dimensioning Problem (RSRDP), we first established the importance of jointly determining which restaurants to offer and how to allocate riders in space and time, demonstrating that demand and supply decisions are fundamentally interlinked. Our numerical experiments showed that an integrated approach consistently outperforms separated methods, leading to higher profitability while still ensuring timely deliveries and balanced workloads. In fulfilling the first objective, we developed a MIP, transformed this to a MILP, incorporating a nested logit framework for customer restaurant choice and combining it with a spatial-temporal network model of rider flows, providing a holistic view of platform operations and clarifying how restaurant availability shapes order volumes and how and where to deploy riders.

Moving toward our second objective, we tested several exact and heuristic solution strategies on datasets of varying sizes. While exact methods were tractable for smaller problems, we found that Benders decomposition and a novel Iterative Assortment Generation (IAG) heuristic effectively scaled up to larger instances. Benders decomposition separated assortment decisions from rider allocation, accelerating solution times by focusing on smaller subproblems, whereas the IAG heuristic incrementally refined the set of candidate assortments until a near-optimal solution emerged. Both approaches performed well across different computational scenarios, reinforcing the value of tailoring solution strategies to the structural properties of the RSRDP rather than relying on traditional off-the-shelf solvers alone.

Having addressed the methodological objectives, we then analyzed how the platform's choices and constraints translate into managerial insights, thereby fulfilling the third objective of evaluating compensation policies and operational decisions in a realistic context. Comparing commission-based (CB) and fixed-employment (FE) policies revealed trade-offs between flexibility, stability, and cost predictability. CB tended to yield higher profits in settings with significant spatial or temporal demand fluctuations, but could also impose more frequent relocations and higher rider turnover. FE provided more predictable labor costs and coverage, yet became expensive when demand spikes required many salaried couriers. Our exploration of parameter extremes, such as short shifts or widely dispersed demand, underscored how multiple factors can converge to accentuate one policy's advantages over the other. In practice, the choice of compensation model will depend on each platform's preference for profitability, cost stability, rider satisfaction, and the potential for demand surges.

Viewed collectively, these findings confirm that an integrated approach to restaurant assortment selection and rider dimensioning is critical for maximizing overall performance. Managing the demand side (through carefully curated assortments) shapes the order distribution in time and space, which in turn allows a more efficient allocation of riders. Platforms adopting this perspective should encourage cross-functional collaboration between marketing and operations teams so that restaurant offerings and rider schedules reinforce each other. They might also consider hybrid pay structures that combine the flexibility of commission-based arrangements with the predictability of guaranteed wages. Furthermore, there is scope to extend this research to rolling-horizon planning, dynamic assortment updates, richer rider heterogeneity, and additional performance metrics such as

rider well-being or customer satisfaction.

In answering the research question, our results demonstrate that the design of restaurant offerings and rider dimensioning is paramount to competing successfully in a crowded meal delivery market. By meeting the three research objectives, we have shown that an integrated optimization framework can substantially improve profitability, especially when supported by robust heuristic and decomposition methods suitable for large-scale instances. The insights and methods presented here apply broadly to on-demand service contexts that similarly depend on interplay between demand-shaping and supply-side orchestration, offering a platform for more adaptive, equitable, and sustainable approaches in the emerging urban logistics landscape.

7.1. Discussion and recommendations

While the proposed model and results offer encouraging evidence that an integrated approach can improve both profitability and operational efficiency, several limitations remain. The parameter settings employed here capture only approximate reflections of real-world conditions. For instance, more detailed information on customer behavior, courier turnover, and geographically specific ordering patterns would likely sharpen quantitative estimates and guide more targeted managerial strategies. Likewise, the scope of validation could be broadened by repeatedly sampling a wider variety of instances, thereby increasing confidence in both the robustness and generalizability of the outcomes. Nevertheless, the unified methodology introduced in this study provides a solid foundation for decision support by incorporating both demand and supply considerations in a single framework.

Looking ahead, several research directions appear particularly promising. One fruitful extension would be to systematically integrate a hybrid compensation model in which some riders are assigned to fixed employment contracts and others to commission-based, potentially optimizing the mix of courier types across zones and time. Such a model could better capture the nuances of real-world labor markets, where different compensation schemes are used simultaneously. Another avenue would involve developing multi-objective formulations that explicitly trade off maximizing platform profit against other performance indicators, such as on-time delivery ratios or rider satisfaction. This could illuminate the compromises managers must make between cost efficiency and service quality.

An additional research priority is to make the design of service districts more endogenous, allowing for variable granularity and consumer segmentation rather than assuming a fixed district map. Finer segmentation, possibly augmented by time-varying assortments that align restaurant offerings with fluctuating demand, might generate higher overall profitability but would also increase modeling complexity. Meanwhile, tracking couriers on an individual basis, converting parts of the problem from a minimum-cost flow paradigm to richer integer-based routing, holds potential for more precise scheduling around travel times and shift boundaries, though it would require larger computational resources and advanced solving techniques.

Moreover, the solution approach itself can be extended in ways that bolster both performance and scalability. One possibility is to adopt a full column generation scheme in which the pricing problem (PP) is solved by a heuristic, then compare this approach directly with the Iterative Assortment Generation (IAG) algorithm to identify trade-offs in quality and runtime. A deeper investigation into how these column-generation heuristics perform under varying instance sizes or cost structures could reveal further gains in solution accuracy or speed. Additional heuristic strategies, along with upper-bound estimation tools, would help practitioners of large-scale systems gauge the gap between a time-feasible solution and any theoretical optimum, particularly relevant in environments where quick responses are paramount.

In conclusion, embedding restaurant assortment choices within rider allocation and testing alternative compensation schemes demonstrates that meal delivery platforms can indeed achieve meaningful improvements in profitability and operational performance. Although several avenues remain open for future research, this work provides both methodological progress and tangible guidance. In an industry characterized by speed, flexibility, and cost competitiveness, seamlessly interlinking menu offerings and fleet management emerges as a powerful means of sustaining growth and success.

References

L. Alfandari, A. Hassanzadeh, and I. Ljubić. An exact method for assortment optimization under the nested logit model. *European Journal of Operational Research*, 291(3):830–845, June 2021. ISSN 03772217. doi: 10.1016/j.ejor.2020.12.007. URL https://linkinghub.elsevier.com/retrieve/pii/S0377221720310171.

G. Amsterdam. Population Density Amsterdam | Website Onderzoek en Statistiek, 2024a. URL https://onderzoek.amsterdam.nl/interactief/dashboard-kerncijfers?tab=indicator&thema=bevolking&indicator=BEVDICHT&indeling=wijken&jaar=2024&gebied=KA&taal=en.

G. Amsterdam. Amsterdam districts and neighborhoods, 2024b. URL https://www.amsterdamsights.com/about/neighborhoods.html.

D. Aparicio, D. Prelec, and W. Zhu. Choice Overload and the Long Tail: Consideration Sets and Purchases in Online Platforms. *Manufacturing & Service Operations Management*, page msom.2021.0318, Jan. 2025. ISSN 1523-4614, 1526-5498. doi: 10.1287/msom.2021.0318. URL https://pubsonline.informs.org/doi/10.1287/msom.2021.0318.

M. G. H. Bell, D. T. Le, J. Bhattacharjya, and G. Geers. On-Demand Meal Delivery: A Markov Model for Circulating Couriers. *Transportation Science*, page trsc.2024.0513, Sept. 2024. ISSN 0041-1655, 1526-5447. doi: 10.1287/trsc.2024.0513. URL https://pubsonline.informs.org/doi/10.1287/trsc.2024.0513.

J. G. Carlsson, S. Liu, N. Salari, and H. Yu. Provably Good Region Partitioning for On-Time Last-Mile Delivery. *SSRN Electronic Journal*, 2021. ISSN 1556-5068. doi: 10.2139/ssrn.3915544. URL https://www.ssrn.com/abstract=3915544.

A. Charnes and W. W. Cooper. An explicit general solution in linear fractional programming. *Naval Research Logistics Quarterly*, 20(3):449–467, Sept. 1973. ISSN 0028-1441, 1931-9193. doi: 10.1002/nav.3800200308. URL https://onlinelibrary.wiley.com/doi/10.1002/nav.3800200308.

B.-L. Chua, S. Karim, S. Lee, and H. Han. Customer Restaurant Choice: An Empirical Analysis of Restaurant Types and Eating-Out Occasions. *International Journal of Environmental Research and Public Health*, 17(17):6276, Aug. 2020. ISSN 1660-4601. doi: 10.3390/ijerph17176276. URL https://www.mdpi.com/1660-4601/17/17/6276.

J. M. Davis, G. Gallego, and H. Topaloglu. Assortment Optimization Under Variants of the Nested Logit Model. *Operations Research*, 62(2):250–273, Apr. 2014. ISSN 0030-364X, 1526-5463. doi: 10.1287/opre.2014.1256. URL https://pubsonline.informs.org/doi/10.1287/opre.2014.1256.

Delft High Performance Computing Centre. DelftBlue Supercomputer (Phase 2). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2, 2024.

P. Fakfare. Influence of service attributes of food delivery application on customers' satisfaction and their behavioural responses: The IPMA approach. *International Journal of Gastronomy and Food Science*, 25:100392, Oct. 2021. ISSN 1878450X. doi: 10.1016/j.ijgfs.2021.100392. URL https://linkinghub.elsevier.com/retrieve/pii/S1878450X21000913.

M. Gansterer, R. F. Hartl, and S. Wieser. Assignment constraints in shared transportation services. *Annals of Operations Research*, 305(1-2):513–539, Oct. 2021. ISSN 0254-5330, 1572-9338. doi: 10.1007/s10479-020-03522-x. URL https://link.springer.com/10.1007/s10479-020-03522-x.

S. R. Kancharla, T. Van Woensel, S. T. Waller, and S. V. Ukkusuri. Meal Delivery Routing Problem with Stochastic Meal Preparation Times and Customer Locations. *Networks and Spatial Economics*, Sept. 2024. ISSN 1566-113X, 1572-9427. doi: 10.1007/s11067-024-09643-1. URL https://link.springer.com/10.1007/s11067-024-09643-1.

J. Ke, C. Wang, and X. Li. Equilibrium Analysis for On-Demand Food Delivery Markets. *SSRN Electronic Journal*, 2022. ISSN 1556-5068. doi: 10.2139/ssrn.4291481. URL https://www.ssrn.com/abstract=4291481.

J. Li, S. Yang, W. Pan, Z. Xu, and B. Wei. Meal delivery routing optimization with order allocation strategy based on transfer stations for instant logistics services. *IET Intelligent Transport Systems*, 16(8):1108–1126, Aug. 2022. ISSN 1751-956X, 1751-9578. doi: 10.1049/itr2.12206. URL https://onlinelibrary.wiley.com/doi/10.1049/itr2.12206.

X. Li, J. Ke, H. Yang, H. Wang, and Y. Zhou. An aggregate matching and pick-up model for mobility-on-demand services. *Transportation Research Part B: Methodological*, 190:103070, Dec. 2024. ISSN 01912615. doi: 10.1016/j.trb.2024.103070. URL https://linkinghub.elsevier.com/retrieve/pii/S0191261524001942.

Z. Li and G. Wang. On-Demand Delivery Platforms and Restaurant Sales. *Management Science*, page mnsc.2021.01010, Oct. 2024. ISSN 0025-1909, 1526-5501. doi: 10.1287/mnsc.2021.01010. URL https://pubsonline.informs.org/doi/10.1287/mnsc.2021.01010.

S. Liu and Z. Luo. On-Demand Delivery from Stores: Dynamic Dispatching and Routing with Random Demand. *Manufacturing & Service Operations Management*, 25(2):595–612, Mar. 2023. ISSN 1523-4614, 1526-5498. doi: 10.1287/msom.2022.1171. URL https://pubsonline.informs.org/doi/10.1287/msom.2022.1171.

S. Liu, L. He, and Z.-J. M. Shen. Data-Driven Order Assignment for Last Mile Delivery. *SSRN Electronic Journal*, 2018. ISSN 1556-5068. doi: 10.2139/ssrn.3179994. URL https://www.ssrn.com/abstract=3179994.

S. Ropke and D. Pisinger. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, 40(4):455–472, Nov. 2006. ISSN 0041-1655, 1526-5447. doi: 10.1287/trsc.1050.0135. URL https://pubsonline.informs.org/doi/10.1287/trsc.1050.0135.

Z. Steever, M. Karwan, and C. Murray. Dynamic courier routing for a food delivery service. *Computers & Operations Research*, 107:173–188, July 2019. ISSN 03050548. doi: 10.1016/j.cor.2019.03.008. URL https://linkinghub.elsevier.com/retrieve/pii/S0305054819300681.

C. S. Tang, J. Bai, K. C. So, X. M. Chen, and H. Wang. Coordinating Supply and Demand on an On-Demand Platform: Price, Wage, and Payout Ratio. *SSRN Electronic Journal*, 2016. ISSN 1556-5068. doi: 10.2139/ssrn.2831794. URL http://www.ssrn.com/abstract=2831794.

M. W. Ulmer and M. Savelsbergh. Workforce Scheduling in the Era of Crowdsourced Delivery. *Transportation Science*, 54(4):1113–1133, July 2020. ISSN 0041-1655, 1526-5447. doi: 10.1287/trsc.2020.0977. URL https://pubsonline.informs.org/doi/10.1287/trsc.2020.0977.

M. W. Ulmer, B. W. Thomas, A. M. Campbell, and N. Woyak. The Restaurant Meal Delivery Problem: Dynamic Pickup and Delivery with Deadlines and Random Ready Times. *Transportation Science*, 55(1):75–100, Jan. 2021. ISSN 0041-1655, 1526-5447. doi: 10.1287/trsc.2020.1000. URL https://pubsonline.informs.org/doi/10.1287/trsc.2020.1000.

G. Xue, Z. Wang, and G. Wang. Optimization of Rider Scheduling for a Food Delivery Service in O2O Business. *Journal of Advanced Transportation*, 2021:1–15, May 2021. ISSN 2042-3195, 0197-6729. doi: 10.1155/2021/5515909. URL https://www.hindawi.com/journals/jat/2021/5515909/.

Y. Yang, S. W. Umboh, and M. Ramezani. Freelance drivers with a decline choice: Dispatch menus in on-demand mobility services for assortment optimization. *Transportation Research Part B: Methodological*, 190:103082, Dec. 2024. ISSN 01912615. doi: 10.1016/j.trb.2024.103082. URL https://linkinghub.elsevier.com/retrieve/pii/S0191261524002066.

B. Yildiz and M. Savelsbergh. Provably High-Quality Solutions for the Meal Delivery Routing Problem. *Transportation Science*, 53(5):1372–1388, Sept. 2019a. ISSN 0041-1655, 1526-5447. doi: 10.1287/trsc.2018.0887. URL https://pubsonline.informs.org/doi/10.1287/trsc.2018.0887.

B. Yildiz and M. Savelsbergh. Service and capacity planning in crowd-sourced delivery. *Transportation Research Part C: Emerging Technologies*, 100:177–199, Mar. 2019b. ISSN 0968090X. doi: 10.1016/j.trc.2019.01.021. URL https://linkinghub.elsevier.com/retrieve/pii/S0968090X18311513.

A

Scientific paper

Restaurant Selection and Rider Dimensioning Problem for Meal Delivery Platforms

Frederiek Backers
Faculty of Technology, Civil Engineering and Geosciences, Delft University of Technology.

Dongyang Xia
Faculty of Civil Engineering and Geosciences, Delft University of Technology.

Shadi Sharif Azadeh
Faculty of Civil Engineering and Geosciences, Delft University of Technology.

Yousef Maknoon
Faculty of Technology, Policy, and Management, Delft University of Technology.

We introduce the Restaurant Selection and Rider Dimensioning Problem (RSRDP) for meal delivery platforms operating in urban areas. The goal is to jointly decide which restaurants to offer and how many riders to deploy in a spatial-temporal network to maximize expected profit, defined as restaurant commission revenues minus rider costs, while ensuring high service quality. Customers arrive to the platform probabilistically, yet decisions on where they order is shaped through restaurant assortment optimization, where each service district curates a subset of restaurants based on a nested logit model capturing customer choices. Meanwhile, rider dimensioning decisions account for spatial and temporal fulfillment of orders, achieving high service quality standards. We compare two compensation schemes: commission-based, where riders are paid per delivery, and fixed employment, where they are paid hourly. Our model is solved using our novel iterative assortment generation algorithm combined with Benders decomposition. Results demonstrate that integrating assortment and rider decisions enhances platform profitability, service quality, and workforce stability.

Key words: Meal Delivery Platform, Rider Dimensioning, Assortment Optimization, Service Design

---

1. Introduction

Meal delivery platforms operate in an uncertain environment, where customers arrive to the system and place their order, which must be fulfilled in a timely manner. These platforms aim to achieve some key quality service level elements: ensuring on-time delivery to maintain customer satisfaction, providing a diverse selection of restaurants to attract demand, and managing operational costs to remain profitable. These elements translate to the challenges these platforms face: the efficient dimensioning of riders, who must be engaged based on their spatial and temporal distribution to meet fluctuating demand, and which restaurants to offer on the platforms so as to maximize expected profit while offering diverse options.

Existing research on meal delivery platforms primarily focuses on either operational efficiency or demand-side management, but rarely integrates both aspects. Prior studies can be grouped into two main categories. The first category focuses on the operational side, optimizing routing costs and reducing delivery times (Xue, Wang, and Wang 2021, Kancharla et al. 2024, Ulmer et al. 2021), offering valuable insights into fleet operations and dispatching strategies. However, these studies generally assume that customer demand is exogenous and fixed, meaning the platform has no control over how demand can be shaped. Other works examine related operational challenges such as delivery time estimation, rider shift scheduling, and courier imbalances (Liu, He, and Shen 2018, Tang et al. 2016). The second category focuses on restaurant selection which is mainly studied in the domain of assortment optimization, aiming for revenue maximization of offering products to customers. These studies, however, neglect the operational implications of assortment decisions on rider dimensioning.

A key insight of this research is that meal delivery platforms function as interconnected systems where each operational decision influences platform performance. Customers expect timely deliveries, while riders require manageable workloads, and platforms must remain profitable. Unlike previous studies that address these elements in isolation, we recognize that the interactions among them create significant research opportunities. While customer arrival is exogenous, platform controlled decision-making, such as restaurant selection and rider dimensioning, and their interaction, are endogenous and can be shaped by designing the services. In this research, we define and present a model for the *Restaurant Selection and Rider Dimensioning Problem* (RSRDP), which jointly determines restaurant selection and rider dimensioning to maximize expected platform profitability while ensuring high service levels. Profit is defined as restaurant commission revenue minus rider costs. We compare two compensation policies for the rider costs: commission-based (CB), where riders get paid per delivery completed, and fixed employment (FE), where riders receive hourly wages. We evaluate these policies to assess their impact on costs, rider availability, and service quality.

To solve the RSRDP, we propose a novel iterative assortment generation heuristic, inspired by column generation, to iteratively generate and evaluate candidate assortments of restaurants to efficiently converge to a good solution. Additionally, we apply Benders decomposition to decouple assortment decisions and rider dimensioning, accelerating convergence in large instances. Our results demonstrate that integrating assortment and rider dimensioning decisions consistently enhances platform profitability while maintaining service quality, including timely deliveries and balanced rider workloads. Furthermore, we evaluate the impact of the two rider compensation policies, finding that the commission-based approach improves operational flexibility but increases rider relocations, whereas the fixed employment scheme ensures stable required workforce.

The remainder of this paper is organized as follows: Section 2 reviews related literature. Section 3 presents the conceptual representation, laying the groundwork for the mathematical formulation in Section 4. Section 5 outlines the solution methodology, detailing the iterative assortment generation algorithm. Section 6 presents computational experiments, validating the model's effectiveness with simulated data, and Section 7 summarizes key findings.

2. Related Literature

Meal delivery platforms have reshaped urban food consumption with on-demand, app-based services. They combine transportation logistics with revenue management, requiring decisions that balance customer satisfaction with profitability. Customer satisfaction depends on timely deliveries, high service quality, and ample restaurant options, while profitability stems from cost-efficiency and strategic pricing. To better understand the complexities of meal delivery platforms, we follow the meal delivery process as depicted in Figure 1, examining the perspectives of the customers, orders, and riders at each stage, along with the associated challenges and how current literature tackles these.

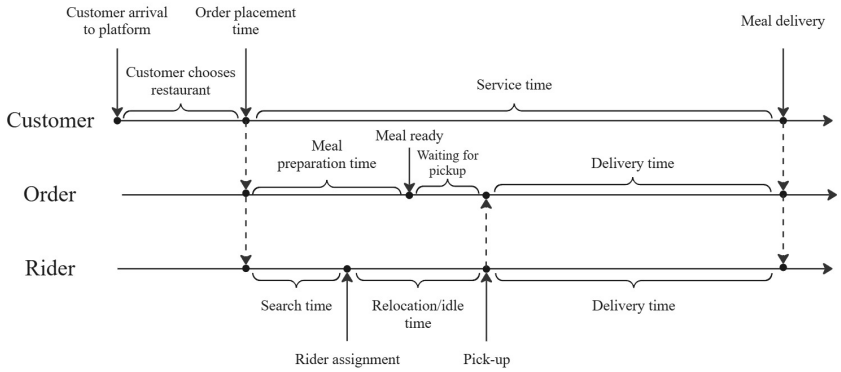

Figure 1 Ordering process timeline from customer, order and rider perspective.

2.1. Request arrival and customer decisions

A primary challenge for meal delivery platforms is managing the dynamic arrival of orders. Customer arrivals are inherently unpredictable, and platforms often model customer arrivals as stochastic processes. For instance, Xue, Wang, and Wang (2021) use an empirical distribution based on real observed data, while Kancharla et al. (2024) adopt a Poisson process. The meal delivery industry shares this challenge with on-demand micro-mobility services. However, unlike services where requests can be declined, meal delivery platforms must fulfill all incoming orders. The study by Li and Wang (2024) incorporates probabilistic demand modeling in mobility-on-demand services, demonstrating its relevance in dynamic dispatching. These approaches align with our work, where we model arrivals of requests probabilistically using a Poisson process.

Once customers are on the platform, they navigate a two-stage decision process. First, based on personal tastes, dietary needs, and even mood, customers select a preferred cuisine category. Next, restaurants offering that cuisine are presented, where the choice is influenced by factors such as estimated delivery time, pricing, and restaurant reputation (Fakfare 2021, Chua et al. 2020). This underscores the importance of curating restaurant offerings that align with consumer tastes, including more heterogeneity customized to the individual, also suggested by Aparicio, Prelec, and Zhu (2025). Our work builds on these insights by integrating customer choice modeling using a nested logit model as presented by Davis, Gallego, and Topaloglu (2014) to aid decision-making on which restaurants to offer.

Once a customer places an order, meal preparation begins at the restaurant. Meal preparation times are uncertain, as they depend on kitchen workload, meal complexity, and restaurant efficiency. This uncertainty can create delays, affecting rider scheduling and customer satisfaction. Ulmer et al. (2021) examine uncertain meal preparation times and propose buffering techniques to minimize delays. Their anticipatory customer assignment approach optimizes real-time order bundling and dispatching. While our model does not explicitly incorporate meal preparation uncertainties, it introduces flexibility in rider assignment to accommodate variability in meal readiness times. This flexibility acts as a buffer for uncertainties in general, ensuring that unexpected delays in preparation do not significantly disrupt delivery operations.

2.2. Rider assignment and supply-demand management

Once an order is placed, matching it with an available rider poses another challenge. The spatial distribution of orders and riders requires quick, efficient dispatch. To address this, Liu, He, and Shen (2018) incorporate predictive travel time analytics into their order assignment model, while Li et al. (2024) improve spatial efficiency using a dynamic matching radius. Additionally, the compensation structure of the riders affects rider availability. Ke, Wang, and Li (2022) explore how different wage schemes impact service quality and profitability, demonstrating that optimized pay schemes influence rider participation and order fulfillment success rates. Rider participation is also studied by Tang et al. (2016), who demonstrate that while higher wages can attract more riders, they might also lead to inefficiencies during low-demand periods. Our research evaluates two static compensation policies, commission-based and fixed employment, to understand their affects on both service quality and profitability.

2.3. Delivery efficiency

After rider assignment, ensuring prompt delivery is key to ensuring service quality. This challenge is addressed in various ways: Liu and Luo (2023) propose a stochastic dynamic driver dispatching system, optimizing routing through Benders decomposition, while Yildiz and Savelsbergh

(2019a) explore multi-objective optimization to balance cost and service quality. Their findings also highlight that compensation schemes and courier schedules play a crucial role in ensuring service reliability. Additionally, Carlsson et al. (2021) investigate how geographic familiarity impacts rider efficiency, proposing a partitioning algorithm to optimize delivery regions. Another upcoming method to increase efficiency includes order bundling within a single trip, Steever, Karwan, and Murray (2019) examine bundling strategies that aim to minimize delays while maintaining routing efficiency. However, Yildiz and Savelsbergh (2019a) analyze the trade-offs between bundling efficiency and service quality, concluding that bundling does not always yield cost savings yet can decrease service quality. Our model does not include bundling but instead focuses on maintaining strict delivery windows to ensure high service quality and timely fulfillment, as assumed in other studies (Ulmer et al. 2021, Kancharla et al. 2024, Li et al. 2022).

Routing efficiency can also be enhanced through strategic rider relocation. Bell et al. (2024) propose a Markov chain-based relocation model where couriers circulate through the city in a structured manner, optimizing transitions based on demand probabilities. Yang, Umboh, and Ramezani (2024) examine how freelance drivers make routing and dispatch decisions based on probabilistic acceptance behavior. While their work focuses on mobility services, it provides insights into how supply-side constraints impact service quality and profitability. In our model, we improve delivery efficiency by strategically relocating couriers based on future demand predictions, ensuring that riders are positioned optimally before orders arrive. Because we can shape demand distribution through assortment optimization of restaurant offerings, we can exert greater control over the spatial distribution of arriving orders.

2.4. Rider scheduling

Beyond real-time rider dispatching, platforms must manage workforce scheduling. Platforms can hire riders as employees or engage them as freelancers. Employed riders provide stability and better workforce planning, but they increase fixed operational costs. Freelance riders offer flexibility, but their availability is uncertain and influenced significantly by compensation structures. Ulmer and Savelsbergh (2020) explore a hybrid workforce model that incorporates both scheduled and unscheduled riders. Their work highlights the importance of structured scheduling while allowing flexibility through crowdsourced labor. The compensation policies implied by the workforce models play a crucial role in rider scheduling. Yildiz and Savelsbergh (2019b) examine how service radius adjustments impact profitability by balancing rider costs and restaurant commissions. They also incorporate restaurant availability as a function of the service radius, allowing the platform to influence demand through spatial adjustments. However, their study assumes exogenous demand and self-scheduling riders who can reject orders, whereas our approach assumes riders must adhere

to platform decision-making on rider movements and incorporates restaurant availability through assortment optimization.

Our work advances the literature by proposing an optimization approach that simultaneously determines the optimal rider dimensioning and the assortment of restaurants being offered to the customer. Additionally, we evaluate static compensation policies and their impact on system efficiency, offering insights into how wage policies influence workforce management and platform profitability. In Section 3, we explain in detail the conceptual representation of this problem.

3. Conceptual representation

This section introduces the notation and core concepts needed for the RSRDP formulation. We structure the problem in a spatial-temporal network and explain the interplay between revenue, costs, customer demand, assortment selection, and rider dimensioning.

3.1. Service design structure

We consider a pre-defined operating area and a nominal day for planning. Our approach employs a two-level spatial representation by dividing this area into larger service districts and smaller hexagonal zones. Service districts, denoted by $d \in D$, capture market heterogeneity by grouping regions with distinct customer preferences and socio-economic traits. These districts also define the curated set of restaurants visible to customers, enabling strategic demand shaping through assortment decisions that directly influence order distribution. To manage rider allocation at a more granular level, the area is further partitioned into hexagonal zones, denoted by $m \in M$. These uniformly distributed hexagons capture rider movement dynamics and allow for accurate travel time computation. Riders travel between zones to complete deliveries and can be strategically relocated to balance supply and demand, thereby minimizing delivery times and operational costs. The interaction between service districts and zones introduces spatial-temporal dependencies that our framework explicitly models, capturing the feedback loop between assortment-driven demand generation and rider management.

We let κ be the travel time between adjacent zone centroids. The travel time between any two zones m and m' is given by $\tau_{mm'}$, which depends on κ and the shortest path distance. It is assumed that intrazonal travel time τ_{mm} equals κ, relating to the maximum travel distance within a zone from the hexagonal structure. Additionally, to discretize the day into manageable intervals, we define $t \in T = \{0, \kappa, ..., T_{max}\}$ where κ also serves as the length of each time period. This synchronized spatial-temporal discretization enables us to track the progression of demand and rider movements over the course of the time horizon. The left side of Figure 2 illustrates the service design structure, with hexagonal zones and colored service districts.

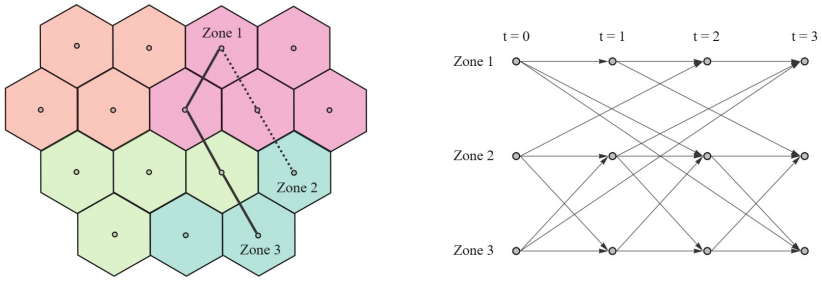

Figure 2 **Hexagonal zone structure with colored service districts (left) and three-zone-four-period spacial-temporal network example where the arcs represent possible courier flows.**

3.2. Rider network activity

We model rider movements through a directed spatial-temporal network $G(N, A)$. Each node $(m, t) \in N$ represents zone $m \in M$ at period $t \in T$. An arc $a = ((m, t), (m', t + \tau_{mm'})) \in A$ indicates that a rider can travel from zone m at time t to zone m' at time $t + \tau_{mm'}$, illustrated in Figure 2 for a three-zone-four-period network. Because all riders are assumed to travel at a constant speed, each arc reflects a possible flow from an earlier time to a later one. The sets of outgoing and incoming arcs at node (m, t) are denoted by $A^+_{(m,t)}$ and $A^-_{(m,t)}$, respectively, and we do not impose capacity constraints on the arcs. Figure 3 presents an example for the origin and destination adjacent arc sets.

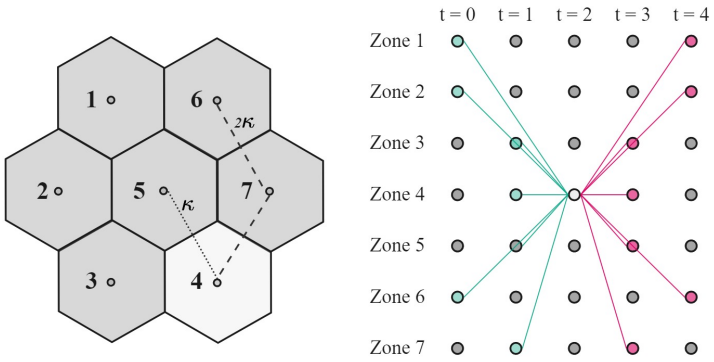

Figure 3 **Example of adjacent nodes for zone 4 structure. Origin arcs $A^-_{(4,2)}$ are presented in blue and destination arcs $A^+_{(4,2)}$ are presented in magenta.**

3.3. Customer arrivals and delivery timeline

Customer arrivals are treated as requests and are aggregated over zones. Specifically, arrivals in zone m at period t denoted λ_{mt} follow a Poisson distribution, capturing the natural fluctuations in demand. We model $\lambda_{mt} \sim Poisson(rate)$, where $rate = Base(t) \cdot (1 + \epsilon_{mt})$. Here, $Base(t)$ is a shape function that rises and falls with typical meal times, and ϵ_{mt} introduces normally distributed

random fluctuations. A request becomes an order if the customer decides to purchase a meal from an available restaurant showcased in the assortment for the district in which the customer is located. How the customer chooses the restaurant is explained in the subsequent section.

Each order placed at time t on the platform must be fulfilled within a delivery window $[t, t+\rho]$ Practically, this window includes meal preparation time and rider travel time, plus a small buffer to accommodate unexpected delays. If η represents the average meal preparation time, and $\tau_{mm'}$ is the travel time from a restaurant located in zone m to the customer's zone m', then the rider must pick up the meal after η and complete the delivery before the total elapsed time reaches ρ. Figure 4 illustrates the possible arcs that may be utilized to satisfy the demand, allowing for some flexibility and possible higher efficiency by providing multiple options in some scenarios.

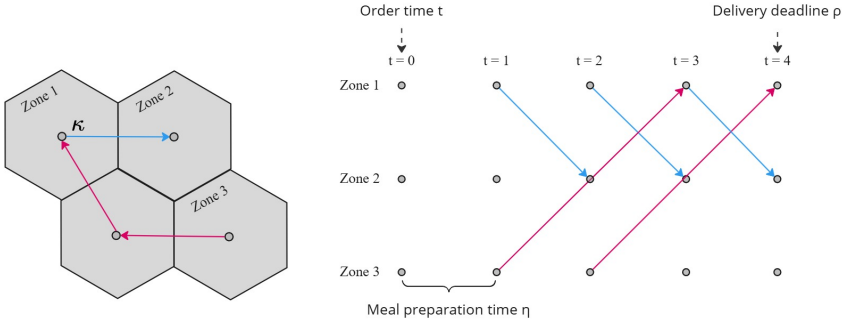

Figure 4 **Illustration of potential courier flow along arcs to meet demand. Orders from Zone 1 to Zone 2 are fulfilled via the blue arcs, while orders from Zone 3 to Zone 1 follow the magenta arcs. Meal preparation time η and the maximum delivery deadline ρ are accounted for.**

3.4. Service district assortment optimization

In addition to managing rider activity, the platform chooses which restaurants to offer in each service district, from which customers can order. Let R be the set of all restaurants in the system, and let $r \in R$. A district d may only include restaurants that can reliably deliver within the delivery window ρ. Specifically, for a customer located in zone m and a restaurant in zone m', $R_m = \{r \in R : \tau_{m'm} \leq \rho - \eta\}$ denotes the set of restaurants capable of serving zone m, and if $b_m^d \in \{0,1\}$ indicates whether zone m is covered by district d, then $R_d = \bigcap_{m \in M : b_m^d = 1} R_m$ is the set of restaurants that can fulfill the delivery window requirements for all zones in district d. Finally, we define a finite set of cuisine types Q, such that $q \in Q$ and each restaurant $r \in R$ belongs to exactly one cuisine type, administered by the parameter $e_r^q \in \{0,1\}$ that is one when restaurant r is of cuisine q. In this way, the set $R_q^d = \bigcap_{m \in M : b_m^d = 1} \{r \in R_m : e_r^q = 1\}$ represents the set of restaurants that may be included for the assortment of district d for cuisine type q. The model may decide which of these restaurants

to offer. We denote the final assortment of restaurants for cuisine type q in service district d by $S_q^d \subseteq R_q^d$.

3.5. Customer purchasing behavior

To model how customers choose restaurants from the platform's assortment, we employ a nested choice framework inspired by Davis, Gallego, and Topaloglu (2014). On meal delivery platforms, customers tend to first choose a cuisine type and then select a restaurant within that category. The nested logit model follows the same structure: customers first choose a nest (cuisine) and thereafter a product within that nest (restaurant). We define v_{qr}^d as the attraction value of restaurant r of cuisine q within district d. This value encapsulates all relevant utility parameters (e.g. price, quality, proximity) and is assumed known from prior choice modeling research or historical data. We also define v_{q0}^d as the attraction of the no-purchase options within cuisine q for district d. Suppose the assortment of cuisine q in district d is $S_q^d \subseteq R_q^d$. The total attraction in nest q is then $V_q^d(S_q^d) = v_{q0}^d + \sum_{r \in S_q^d} v_{qr}^d$. Under the framework, the probability that a customer orders from restaurant $r \in S_q^d$ given they have selected nest q is:

$$\mathbb{P}_{r|q}^d(S_q^d) = \frac{v_{qr}^d}{v_{q0}^d + \sum_{r \in S_q^d} v_{qr}^d} = \frac{v_{qr}^d}{V_q^d(S_q^d)} \tag{3.1}$$

Let p_{qr}^d represent the revenue from an order at restaurant r in cuisine nest q for district d. The expected revenue from customers ordering within nest q is:

$$\pi_q^d(S_q^d) = \frac{\sum_{r \in S_q^d} p_{qr}^d v_{qr}^d}{V_q^d(S_q^d)} = \sum_{r \in S_q^d} p_{qr}^d \cdot \mathbb{P}_{r|q}^d(S_q^d) \tag{3.2}$$

Each cuisine nest q has a dissimilarity parameter $\gamma_q^d \geq 0$, which accounts for the degree of dissimilarity of the restaurants within the nest. We assume that these parameters are also researched a priori. Let v_0^d denote the attraction value of the no-purchase option for choosing any of the nests in district d. If we offer assignments $(S_1^d, ..., S_{|Q|}^d)$ over all nests with $S_q^d \subseteq R_q^d \ \forall q \in Q, \ d \in D$, then a customer chooses nest q in district d with probability:

$$\mathbb{P}_q^d(S_q^d) = \frac{V_q^d(S_q^d)^{\gamma_q^d}}{v_0^d + \sum_{q \in Q} V_q^d(S_q^d)^{\gamma_q^d}} \tag{3.3}$$

Then the probability of choosing restaurant r of cuisine q in district d is given by:

$$\mathbb{P}_{qr}^d(S_q^d) = \mathbb{P}_q^d(S_q^d) \cdot \mathbb{P}_{r|q}^d(S_q^d) = \left(\frac{V_q^d(S_q^d)^{\gamma_q^d}}{v_0^d + \sum_{q \in Q} V_q^d(S_q^d)^{\gamma_q^d}} \right) \cdot \left(\frac{v_{qr}^d}{V_q^d(S_q^d)} \right) = \frac{v_{qr}^d \cdot V_q^d(S_q^d)^{\gamma_q^d - 1}}{v_0^d + \sum_{q \in Q} V_q^d(S_q^d)^{\gamma_q^d}} \tag{3.4}$$

And the expected revenue from customers ordering within district d is:

$$\Pi^d(S_1^d, ..., S_{|Q|}^d) = \sum_{q \in Q} \mathbb{P}_q^d(S_q) \pi_q^d(S_q^d) = \frac{\sum_{q \in Q} \pi_q^d(S_q^d) V_q^d(S_q^d)^{\gamma_q^d}}{v_0^d + \sum_{q \in Q} V_q^d(S_q^d)^{\gamma_q^d}} \tag{3.5}$$

By modeling customer choices in this manner, we can estimate the likelihood of each restaurant being selected by customers located in each service district, which subsequently informs the effective flow of orders in the network, which we relate to the number of needed riders to deliver these orders within the network. Equation 3.5 highlights the independence of assortments across service districts. The only interaction between service districts in this problem arises from riders needed to deliver orders between different zones belonging to different districts. The assortment offered in each district dictates the available restaurants, which in turn influences the customer demand generated at those restaurants.

3.6. Rider dimensioning

Using the customer behavioral probabilities derived, we can estimate the expected number of orders between zones. For each period t and zone m, we determine how many orders should be delivered from zone m' to m and therefore how many couriers are required. Suppose that the set $\hat{R}_m$ denotes the restaurants located in zone m. Given the number of requests λ_{mt} and the probability of choosing restaurant r of cuisine q in district d given by Equation 3.4, we denote the number of orders from restaurants located in zone m to customers located in zone m' starting at time t by $\delta^t_{mm'}$. Then:

$$\delta^t_{mm'} = \sum_{d \in D} \sum_{q \in Q} \sum_{r \in \hat{R}_m \cap R^d_q} \lambda_{m't} \cdot \mathbb{P}^d_{qr}(S^d_q) \tag{3.6}$$

As the calculation incorporates probabilities, the number of orders may result in fractional values, representing an average demand across zones. While we assume that each courier can deliver only one order at a time, allowing fractional courier flows in this tactical decision-making model is not only practical but also analytically beneficial. Let the decision variables $u^{in}_{(m,t)}$ and $u^{out}_{(m,t)}$ represent the number of couriers entering and leaving the system node (m,t) in the spatial-temporal network respectively, essentially functioning as source and sink nodes for the couriers. To determine the total fractional number of couriers needed, we sum $u^{in}_{(m,t)}$ across all nodes: $\sum_{(m,t) \in N} u^{in}_{(m,t)}$. Couriers are permitted to enter the system at node, providing the flexibility needed to optimally meet varying demand patterns. However, we impose constraints to ensure that, on average, couriers work for at least a minimum shift duration θ_{min} and do not exceed a maximum shift duration θ_{max}.

3.7. Profit structure

The objective is to maximize the platform's total expected profit, defined as the difference between revenue and costs. Revenue is obtained through commission paid by restaurants on each order made by customers, based on a percentage of the order value. Each order placed by a customer therefore corresponds to a revenue, which can vary depending on the restaurant r and cuisine q and district d. We denote this revenue by p^d_{qr}.

Costs are related to riders and consist of two parts: expenses related to the compensation of riders, and the expenses related to overhead costs depending on the number of couriers that are used in the system. We consider two compensation policies to compare and test the influence of different policies on the performance of the system. The first policy is defined as Commission Based (CB), where riders are compensated for each order they deliver. We define this cost as c^d_{qr}, similar to the profit structure. Similarly, the expected compensation costs of delivering an order within district d related to the CB policy can be found using Equations 3.2 and 3.5 and replacing p^d_{qr} with c^d_{qr}:

$$\hat{c}^{comp}_{CB} = \frac{\sum_{q\in Q}\sum_{r\in S^d_q} c^d_{qr}\cdot \mathbb{P}^d_{r|q}(S^d_q)V^d_q(S^d_q)^{\gamma^d_q}}{v^d_0 + \sum_{q\in Q} V^d_q(S^d_q)^{\gamma^d_q}} \tag{3.7}$$

The second policy is defined as Fixed Employment (FE), where riders are hired by the platform and get compensated a fixed wage per hour. Let c^t be the discretized wage per time unit for a rider. Then traveling along an arc $a = ((m,t),(m',t+\tau_{mm'})) \in A$ incurs a cost $c_a = c^t \cdot \tau_{mm'}$. Summing this over all arcs provides the total wage payout. We generalize the compensation costs for the policies to $c^{compensation}_{policy}$.

The overhead costs exist for both policies, however the overhead costs related to the FE policy are bigger than for the CB policy, as naturally there are higher costs incurred when hiring riders. We define these costs as $c^{overhead}_{policy}$ per rider such that $policy \in \{CB, FE\}$ and calculate the costs based on the number of incoming couriers within the spatial-temporal network:

$$\hat{c}^{overhead}_{policy} = c^{overhead}_{policy} \cdot \sum_{(m,t)\in N} u^{in}_{(m,t)} \tag{3.8}$$

The conceptual framework presented in this section underpins the decision-making mechanisms of the mathematical formulation presented in the next section.

4. Mathematical formulation

We propose a mathematical formulation for the RSRDP that enumerates all possible assortments for each service district d and cuisine nest q. For each (d,q) pair, we introduce binary decision variable $z_{S^d_q} \in \{0,1\}$ to indicate whether a particular assortment $S^d_q \subseteq R^d_q$ of restaurants is offered. The continuous variable w_a represents the rider flow on arc $a \in A$, while $u^{in}_{(m,t)}$ and $u^{out}_{(m,t)}$ represent the incoming and exiting riders at node $(m,t) \in N$. The sets and parameters used in the formulation are summarized in Appendix A, and we refer the reader to the conceptual framework (Section 3) for further details on the problem setting. We first present the mathematical model in its nonlinear

form. The linearization of this formulation is presented in Section 4.1. The problem is presented as follows:

$$\begin{aligned} \max \quad & \sum_{d\in D} \mathbb{E}^{rev}_d(z_{S^d_q}) - \mathbb{E}^{cost}_{policy}(w_a, u^{in}_{(m,t)}, u^{out}_{(m,t)}) \\ \text{s.t.} \quad & (4.6) - (4.15) \end{aligned} \tag{4.1}$$

The objective function (4.1) calculates the difference between the expected revenue obtained from orders ordered within all districts based on assortments S^d_q offered and the expected cost from operating riders. In the remainder of this section, we explain each profit component and the constraints.

Expected revenue. We first define the expected revenue contribution from district d. Each district's revenue depends on the number of arriving requests in all zones covered by the district over the time horizon, $\sum_{m\in M}\sum_{t\in T} b^d_m \lambda_{mt}$, and the expected revenue from customers ordering within district d from Equation (3.5), by multiplying these, we obtain the total expected revenue from district d over all arriving requests. To incorporate this we change Equation (3.5) to include the model's decision variable $z_{S^d_q}$:

$$\mathbb{E}^{rev}(z_{S^d_q}) = \frac{\sum_{q\in Q}\sum_{S^d_q\subseteq R^d_q} V^d_q(S^d_q)^{\gamma^d_q}\pi^d_q(S^d_q)\cdot z_{S^d_q}}{v^d_0 + \sum_{q\in Q}\sum_{S^d_q\subseteq R^d_q} V^d_q(S^d_q)^{\gamma^d_q}\cdot z_{S^d_q}} \cdot \sum_{m\in M}\sum_{t\in T} b^d_m\lambda_{mt} \tag{4.2}$$

By summing over all districts one can obtain the total expected revenue.

Expected costs. The expected rider costs depend on the policy implemented, consisting of the compensation costs and the overhead costs. The compensation costs can be defined similarly as the expected profit for the CB policy, for the FE policy we sum the rider flows over all arcs. This gives us the following:

$$\mathbb{E}^{cost}_{policy}(w_a, u^{in}_{(m,t)}, u^{out}_{(m,t)}) = \hat{c}^{compensation}_{policy} + \hat{c}^{overhead}_{policy} \tag{4.3}$$

Where $\hat{c}^{overhead}_{policy}$ is as defined in subsection 3.7 and $\hat{c}^{compensation}_{policy}$ is defined as:

$$\hat{c}^{compensation}_{policy} = \begin{cases} \dfrac{\sum_{q\in Q}\sum_{S^d_q\subseteq R^d_q} V^d_q(S^d_q)^{\gamma^d_q}\sum_{r\in S^d_q} c^d_{qr}\cdot \mathbb{P}^d_{r|q}(S^d_q)\cdot z_{S^d_q}}{v^d_0+\sum_{q\in Q}\sum_{S^d_q\subseteq R^d_q} V^d_q(S^d_q)^{\gamma^d_q}\cdot z_{S^d_q}} \cdot \sum_{m\in M}\sum_{t\in T} b^d_m\lambda_{mt} & \text{if policy = CB} \\ \sum_{a\in A} c_a\cdot w_a & \text{if policy = FE} \end{cases} \tag{4.4}$$

Demand satisfaction constraints. To ensure that enough rider capacity is available to fulfill all customer orders within the allowable delivery window ρ, recall that an order placed at time t at a restaurant located in zone m by a customer located in zone m' can be picked-up in zone m and delivered in zone m' anywhere within the periods $\{t+\eta ..., t+\rho\}$. The number of related orders was

presented by Equation (3.6), and in a similar fashion as for the objective we incorporate decision variable $z_{S_q^d}$ to obtain the number of orders, i.e. demand as specified from restaurant to customer, from zone m to m' ordered at period t:

$$\Delta^t_{mm'} = \lambda_{m't} \cdot \frac{\sum_{q\in Q}\sum_{S_q^d\subseteq R_q^d}\sum_{r\in\hat{R}_m\cap S_q^d} v_{qr}^d \cdot V_q^d(S_q^d)^{\gamma_q^d-1}\cdot z_{S_q^d}^d}{v_0^d + \sum_{q\in Q}\sum_{S_q^d\subseteq R_q^d} V_q^d(S_q^d)^{\gamma_q^d}\cdot z_{S_q^d}} \tag{4.5}$$

Recall that the binary parameter b_m^d equals one when zone m is covered by district d and zero otherwise. The number of riders needed to deliver the orders between any two zones over all districts within the delivery window should therefore be at least $\Delta^t_{mm'}$ if m' is covered by d, summed over all districts d, presented in Constraint (4.6). This constraint may include an overlap in demand generated within the delivery window, e.g. when two orders are placed from zone m adjacent to m' at time t_1 and t_2 respectively, and need to be delivered within delivery windows $[t_1, t_2, t_3]$ and $[t_2, t_3, t_4]$ respectively, the constraint will hold if one rider delivers one order from zone m to m' starting at time t_2. Therefore we also need to add a global constraint on total riders versus total demand, presented in Constraint (4.7). Both constraints include larger or equal signs because couriers may also relocate instead of delivering, resulting in larger flow values, but because rider costs are minimized the total flow of couriers will be minimized as well.

$$\sum_{\substack{t'=t+\tau_{mm'}\\ t+\eta\le t'\le t+\rho}} w_{(m,t)(m',t')} \ge \sum_{d\in D} b^d_{m'}\Delta^t_{mm'} \qquad \forall m,m'\in M,\quad t\in T\setminus\{T_{max}-\rho-\eta,...,T_{max}\} \tag{4.6}$$

$$\sum_{a\in A} w_a \ge \sum_{m\in M}\sum_{m'\in M}\sum_{t\in T}\sum_{d\in D} b^d_{m'}\Delta^t_{mm'} \tag{4.7}$$

$$w_a \in \mathbb{R}_+ \qquad \forall a\in A \tag{4.8}$$

Rider flow constraints. Rider flows are captured via the node-balance constraints and the definitions of $u^{in}_{(m,t)}$ and $u^{out}_{(m,t)}$. For each node $(m,t)\in N$, we require the total incoming flow plus any new riders entering to equal the total outgoing flow plus any riders leaving, presented in Constraint (4.9). Since the total number of riders entering the system must be the same as the total number exiting, we add Constraint (4.10). Finally, we represent operational rules that constrain how long each rider can work on average using the shift duration $[\theta_{min}, \theta_{max}]$ in Constraint (4.11).

$$\sum_{a\in A^-_{(m,t)}} w_a + u^{in}_{(m,t)} = \sum_{a\in A^+_{(m,t)}} w_a + u^{out}_{(m,t)} \qquad \forall (m,t)\in N \tag{4.9}$$

$$\sum_{(m,t)\in N} u^{in}_{(m,t)} = \sum_{(m,t)\in N} u^{out}_{(m,t)} \tag{4.10}$$

$$\theta_{\min}\cdot\sum_{(m,t)\in N} u^{in}_{(m,t)} \le \sum_{a\in A} w_a \le \theta_{\max}\cdot\sum_{(m,t)\in N} u^{in}_{(m,t)} \tag{4.11}$$

$$w_a \in \mathbb{R}_+ \qquad \forall a\in A \tag{4.12}$$

$$u^{in}_{(m,t)}\in\mathbb{R}_+,\quad u^{out}_{(m,t)}\in\mathbb{R}_+ \qquad \forall m\in M,\quad t\in T \tag{4.13}$$

Assortment constraints. Constraint (4.14) guarantees that the model selects precisely one subset S_q^d out of all possible subsets of R_q^d for each (d, q) pair.

$$\sum_{S_q^d \subseteq R_q^d} z_{S_q^d} = 1 \qquad \forall q \in Q, \quad d \in D \tag{4.14}$$

$$z_{S_q^d} \in \{0,1\} \qquad \forall q \in Q, \quad d \in D, \quad S_q^d \subseteq R_q^d \tag{4.15}$$

In the following subsection, we discuss the linearization of the above-mentioned nonlinear constraints and the objective function.

4.1. Linearization

The model presented by (4.1)-(4.15) is non-linear due to constraint (4.6), (4.7) and the objective function (4.1). We follow a two-step procedure to transform these expressions into a Mixed-Integer Linear Program (MILP). We first isolate the fractional expressions in the objective and constraints. For district d, let continuous variable y_d capture the fraction

$$y_d = \frac{\sum_{q \in Q} \sum_{S_q^d \subseteq R_q^d} V_q^d(S_q^d)^{\gamma_q^d} \pi_q^d(S_q^d) \cdot z_{S_q^d}}{v_0^d + \sum_{q \in Q} \sum_{S_q^d \subseteq R_q^d} V_q^d(S_q^d)^{\gamma_q^d} \cdot z_{S_q^d}} = \frac{f_d}{g_d} \tag{4.16}$$

Similarly, for each zone m and district d, define continuous variable x_{md} to represent the fraction

$$x_{md} = \frac{\sum_{q \in Q} \sum_{S_q^d \subseteq R_q^d} \sum_{r \in \hat{R}_m \cap S_q^d} v_{qr}^d \cdot V_q^d(S_q^d)^{\gamma_q^d - 1} \cdot z_{S_q^d}^d}{v_0^d + \sum_{q \in Q} \sum_{S_q^d \subseteq R_q^d} V_q^d(S_q^d)^{\gamma_q^d} \cdot z_{S_q^d}} = \frac{h_{md}}{g_d} \tag{4.17}$$

Next we rewrite each fraction as a product of the new variable and a linear function of $z_{S_q^d}$, $g_d y_d = f_d \ \forall d \in D$ and $g_d x_{md} = h_{md} \quad \forall d \in D, \quad m \in M$, and introduce bounds for the newly defined variables. As all the relevant components of the model are larger or equal to zero, the lower bound trivially becomes zero for all variables. For the upper bound we want to find the maximum values these variables can attain. For the upper bound of $y_d \in [0, y_d^U]$, we want to find $y_d^U = \max y_d = \max\{\frac{f_d}{g_d}\}$. f_d can be maximized by noting that $z_{S_q^d} = 1$ for exactly one $S_q^d \ \ \forall q \in Q, \ d \in D$, therefore $\max\{f_d\} = \sum_{q \in Q} \max_{S_q^d \subseteq R_q^d} V_q^d(S_q^d)^{\gamma_q^d} \pi_q^d(S_q^d)$. Similarly we can show that to minimize g_d we get $\min g_d = v_0^d + \sum_{q \in Q} \min_{S_q^d \subseteq R_q^d} V_q^d(S_q^d)^{\gamma_q^d}$. Combining these results we get $y_d^U = \frac{\max_z f_d}{\min_z g_d} = \frac{\sum_{q \in Q} \max_{S_q^d \subseteq R_q^d} V_q^d(S_q^d)^{\gamma_q^d} \pi_q^d(S_q^d)}{v_0^d + \sum_{q \in Q} \min_{S_q^d \subseteq R_q^d} V_q^d(S_q^d)^{\gamma_q^d}}$. Similarly, we find the upper bound for $x_{md} \in [0, x_{md}^U]$. Combining all these results we obtain the following formulation:

$$\max \quad \sum_{d \in D} y_d \cdot \sum_{m \in M} \sum_{t \in T} b_m^d \lambda_{mt} - \mathbb{E}_{policy}^{cost}(w_a, u_{(m,t)}^{in}, u_{(m,t)}^{out}) \tag{4.18}$$

$$\text{s.t.} \quad (4.8) - (4.15)$$

$$g_d y_d = f_d \qquad \forall d \in D \tag{4.19}$$

$$g_d x_{md} = h_{md} \qquad \forall d \in D, \quad m \in M \tag{4.20}$$

$$\sum_{\substack{t'=t+\tau_{mm'} \\ t+\eta \le t' \le t+\rho}} w_{(m,t)(m',t')} \ge \lambda_{m't} \sum_{d \in D} b^d_{m'} x_{md} \tag{4.21}$$

$$\forall m, m' \in M,\ t \in T \setminus \{T_{max} - \rho - \eta, ..., T_{max}\}$$

$$\sum_{a \in A} w_a \ge \sum_{m \in M} \sum_{m' \in M} \sum_{t \in T} \lambda_{m't} \sum_{d \in D} b^d_{m'} x_{md} \tag{4.22}$$

$$y_d, f_d, g_d \in [0, y^U_d], \quad x_{md}, h_{md} \in [0, x^U_{md}] \qquad \forall d \in D, \quad m \in M \tag{4.23}$$

Note that for the CB policy we can similarly reformulate the fractional term. Yet still, constraints (4.19) and (4.20) are bilinear in terms of decision variables $z_{S^d_q}$, y_d, g_d and x_{md}. We can further linearize these constraints by applying the commonly used linearization technique by Charnes and Cooper (1973), resulting in adding additional constraints to the model. We introduce auxiliary continuous variables $l_{S^d_q}$ and $k^m_{S^d_q}$ such that $l_{S^d_q} = z_{S^d_q} \cdot y_d$ and $k^m_{S^d_q} = z_{S^d_q} \cdot x_{md}$. We then add the following linearization constraints:

$$l_{S^d_q} \le y_d \qquad \forall d \in D, \quad q \in Q, \quad S^d_q \subseteq R^d_q \tag{4.24}$$

$$l_{S^d_q} \le y^U_d \cdot z_{S^d_q} \qquad \forall d \in D, \quad q \in Q, \quad S^d_q \subseteq R^d_q \tag{4.25}$$

$$l_{S^d_q} \ge y_d - y^U_d \cdot (1 - z_{S^d_q}) \qquad \forall d \in D, \quad q \in Q, \quad S^d_q \subseteq R^d_q \tag{4.26}$$

$$k^m_{S^d_q} \le x_{md} \qquad \forall d \in D, \quad q \in Q, \quad S^d_q \subseteq R^d_q, \quad m \in M \tag{4.27}$$

$$k^m_{S^d_q} \le x^U_{md} \cdot z_{S^d_q} \qquad \forall d \in D, \quad q \in Q, \quad S^d_q \subseteq R^d_q, \quad m \in M \tag{4.28}$$

$$k^m_{S^d_q} \ge x_{md} - x^U_{md} \cdot (1 - z_{S^d_q}) \qquad \forall d \in D, \quad q \in Q, \quad S^d_q \subseteq R^d_q, \quad m \in M \tag{4.29}$$

$$l_{S^d_q} \in [0, y^U_d], \quad k^m_{S^d_q} \in [0, x^U_{md}] \qquad \forall d \in D, \quad q \in Q, \quad S^d_q \subseteq R^d_q, \quad m \in M \tag{4.30}$$

By incorporating the new auxiliary variables into the formulation the bilinear constraints (4.19) and (4.20) respectively change to:

$$\sum_{q \in Q} \sum_{S^d_q \subseteq R^d_q} V^d_q (S^d_q)^{\gamma^d_q} \pi^d_q (S^d_q) \cdot z_{S^d_q} = v^d_0 \cdot y_d + \sum_{q \in Q} \sum_{S^d_q \subseteq R^d_q} V^d_q (S^d_q)^{\gamma^d_q} \cdot l_{S^d_q} \qquad \forall d \in D \tag{4.31}$$

$$\sum_{q \in Q} \sum_{S^d_q \subseteq R^d_q} \sum_{r \in \hat{R}_m \cap S^d_q} v^d_{qr} \cdot V^d_q (S^d_q)^{\gamma^d_q - 1} \cdot z_{S^d_q} = v^d_0 \cdot x_{md} + \sum_{q \in Q} \sum_{S^d_q \subseteq R^d_q} V^d_q (S^d_q)^{\gamma^d_q} \cdot k^m_{S^d_q} \qquad \forall d \in D, \quad m \in M \tag{4.32}$$

These linearizations replace all the bilinear terms the model, allowing it to be formulated as a MILP, which can be effectively solved by a typical branch-and-bound solver like Gurobi, able of reaching global optimality. However, the problem grows exponentially in variables due to the number of possible combinations of restaurants for the assortments. Therefore, we need to carefully search the solution space for which we present our resolution approach in Section 5.

5. Resolution approach

In addressing the problem, we propose a novel solution methodology defined as the Iterative Assortment Generation (IAG) algorithm that incorporates Benders decomposition. This approach is designed to balance computational efficiency with solution quality, and it leverages problem-specific insights to navigate the vast combinatorial space of potential restaurant assortments. The proposed methodology is evaluated through numerical experiments to assess performance in terms of solution quality and computational efficiency, presented in Section 6.

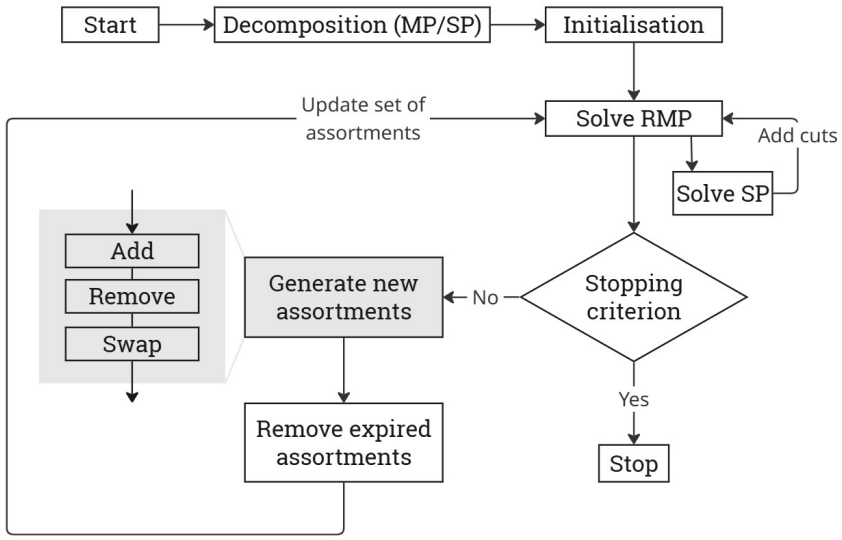

Figure 5 Overview diagram of the IAG algorithm.

Figure 5 summarizes the overall methodology. The IAG algorithm starts with decomposing the problem into a master and subproblem as described in subsection 5.1. Next, the initial set of assortments that form the Restricted Master Problem (RMP) is generated, explained in subsection 5.2. The RMP is solved using a standard solver such as `Gurobi`. The routing subproblem is solved to generate dual information that produces Benders cuts, which are integrated back into the RMP. Concurrently, the IAG algorithm refines the assortment set via heuristic operators and a removal threshold, and the process iterates until a stopping criterion is met, which we define as a time limit or maximum iteration count.

5.1. Benders decomposition

Benders decomposition is employed to decompose the problem and solve it more efficiently: the master problem (MP) on assortment selection to maximize revenue and the rider subproblem (SP) to minimize rider costs. In our formulation, the master problem is an integer program that selects

an assortment for each service district and cuisine nest, while the subproblem is a continuous optimization model that determines the allocation of riders throughout the system based on the chosen assortments. Mathematically, for the RSRDP formulated in Section 4, the Benders decomposition formulation is the following:

Master Problem (MP).

$$\max \quad \sum_{d\in D} y_d \cdot \sum_{m\in M}\sum_{t\in T} b_m^d \lambda_{mt} - \Theta \tag{5.1}$$

$$\text{s.t.} \quad \sum_{S_q^d \in \mathcal{B}_q^d} z_{S_q^d} = 1 \qquad \forall q \in Q, \quad d \in D \tag{5.2}$$

$$(4.24) - (4.32)$$

$$\Theta \geq Z + \sum_{d\in D}\sum_{m\in M} \phi_{md} \cdot (x_{md} - \hat{x}_{md}) \tag{5.3}$$

Constraint (5.3) represents the Bender's cut from the subproblem, where ϕ_{md} is the dual variable of constraint (5.7) in the subproblem. Given a solution $\hat{x}_{md}$ from the MP, the subproblem is presented.

Subproblem (SP).

$$\min \quad Z \;=\; \mathbb{E}^{cost}_{policy}(w_a, u^{in}_{(m,t)}, u^{out}_{(m,t)}) \tag{5.4}$$

$$\text{s.t.} \quad (4.9) - (4.13)$$

$$\sum_{\substack{t'=t+\tau_{mm'} \\ t+\eta\leq t'\leq t+\rho}} w_{(m,t)(m',t')} \geq \lambda_{m't} \sum_{d\in D} b_{m'}^d x_{md} \quad \forall m, m' \in M,\ t \in T \setminus \{T_{max} - \rho - \eta, ..., T_{max}\} \tag{5.5}$$

$$\sum_{a\in A} w_a \geq \sum_{m\in M}\sum_{m'\in M}\sum_{t\in T} \lambda_{m't} \sum_{d\in D} b_{m'}^d x_{md} \tag{5.6}$$

$$x_{md} = \hat{x}^*_{md} \qquad \forall m \in M, \quad d \in D \tag{5.7}$$

The optimal vector $\hat{x}_{md}$ represents the demand generated by the assortment plan, obtained by solving the master problem. The resulting cut from the SP is subsequently added to the MP, which is a process that repeats itself until the MP reaches optimality. By incorporating Benders optimality cuts into the MP, the formulation is progressively tightened, enhancing convergence to the optimal solution.

5.2. Iterative Assortment Generation Algorithm

Our initial exploration focused on an exact solution framework employing column generation combined with a branch-and-price method augmented with Benders decomposition. In this exact framework, the master problem contains the assortment selection variables and is iteratively tightened by incorporating Benders cuts derived from the routing subproblem. Unfortunately, the non-convexity of the associated pricing problem led to intractability. To address this, we develop the Iterative

Assortment Generation (IAG) algorithm described in Algorithm 1, inspired by the column generation approach, which is a heuristic that leverages the structure of the problem without relying on dual-based pricing.

At the heart of the IAG algorithm are three heuristic operators (`add`, `remove`, and `swap`) that generate new assortments by exploring the neighborhood of the current best solution. These operators function similarly to the pricing step in column generation; however, they are tailored to our problem structure and do not depend on dual variables. This allows us to maintain integrality throughout the iterative process, ensuring that every generated solution is feasible in the original problem space. Algorithm 1 presents the pseudo-code of the IAG algorithm. Each component of the algorithm is explained in more detail in the subsequent subsections.

Algorithm 1: Iterative Assortment Generation Algorithm

Input: Data instance, time limit T_{limit}, maximum number of iterations I_{limit}, inactivity threshold φ, last used iteration $\mathcal{I}_{\text{last}}$.

Output: Best objective P^* and associated assortments $\mathcal{A}^*$.

Initialize: $P^* \leftarrow$ NBR objective, $\mathcal{A}^* \leftarrow$ NBR assortments, iteration counter $i \leftarrow 0$, $\mathcal{I}_{\text{last}}(A^*) \leftarrow 0$.

while $i < I_{limit}$ ***and*** *runtime* $< T_{limit}$ **do**

- Solve the **Restricted Master Problem (RMP)** using the Benders decomposition method;
- Extract selected assortments $\mathcal{A}_i$ and objective P_i
- **if** $P_i > P^*$ **then**
  - Update $P^* \leftarrow P_i$, $\mathcal{A}^* \leftarrow \mathcal{A}_i$
- Update $\mathcal{I}_{\text{last}}(A^*) \leftarrow i$;
- Generate new assortments based on selected assortments and add to RP;
- `// Section 5.2.2: generation`
- Remove unused assortments based on inactivity threshold and remove from RP;
- `// Section 5.2.3: removal`
- Increment iteration counter $i \leftarrow i + 1$

return P^* *and* $\mathcal{A}^*$

5.2.1. Initial Nested-By-Revenue assortments. Under the assumptions that the within-nest no-purchase option satisfies $v_{q0}^d = 0$ and that the dissimilarity parameter $\gamma_q^d \leq 1$ for all $d \in D,\ q \in Q$, Davis, Gallego, and Topaloglu (2014) showed that the optimal assortment can be constructed by considering only the nested-by-revenue (NBR) assortments. For each cuisine nest, restaurants are indexed in descending order of revenue, and the optimal assortment is obtained by including restaurants sequentially until the marginal revenue becomes non-positive. These NBR solutions for each district-cuisine pair (d, q) are used as the initial candidates in our IAG algorithm.

5.2.2. Assortment generation. The pseudo-code for generating new assortments in the IAG framework is presented in Algorithm 2. It focuses on refining the set of possible assortments ($\mathcal{A}_{poss}$) by leveraging modifications to previously optimal selected assortments ($\mathcal{A}_{prev}$). At each iteration, the algorithm evaluates restaurants within the previously optimal assortment for a specific cuisine and district combination (d, q) ($\mathcal{S}_{prev}$) and attempts to apply the three key operators to create new candidate assortments:

- `add`**:** For a given current assortment, select a restaurant not included in the set based on its profit contribution (i.e., the product of its revenue p^d_{qr} and attraction value v^d_{qr}). A weighted random selection favors higher-profit restaurants, prioritizing those with greater expected profitability.
- `remove`**:** For a given current assortment, eliminate a restaurant from the assortment based on its inverse profit contribution. Lower-profit restaurants are more likely to be removed.
- `swap`**:** Replace a restaurant in the current assortment with one not present in it, using the weighted selection rules from the `add` and `remove` operators.

A record is maintained to ensure only unique assortments are generated. If no further unique assortments can be created, the algorithm halts the generation process.

Algorithm 2: Generate New Assortments for Selected Assortments

Input: Data instance, previous optimal selected assortments $\mathcal{A}_{\text{prev}}$, current possible assortments $\mathcal{A}_{\text{poss}}$, set R^d_q.

Output: Updated possible assortments $\mathcal{A}'_{\text{poss}}$.

foreach $(d, q) \in \mathcal{A}_{prev}$ **do**

- Extract previously optimal selected assortment $\mathcal{S}_{\text{prev}}$;
- Calculate scores for each restaurant in R^d_q based on p^d_{qr} and v^d_{qr};
- Attempt to `add`: Select a restaurant not in $\mathcal{S}_{\text{prev}}$ and add it;
- `// Section: add`
- Attempt to `remove`: Remove a restaurant from $\mathcal{S}_{\text{prev}}$;
- `// Section: remove`
- Attempt to `swap`: Swap a restaurant in $\mathcal{S}_{\text{prev}}$ with one not in it;
- `// Section: swap`
- Add new assortments to $\mathcal{A}'_{\text{poss}}$

return $\mathcal{A}'_{poss}$

5.2.3. Assortment removal. The assortment removal pseudo-code in Algorithm 3 ensures that the set of available assortments remains efficient by removing outdated options. Each assortment tracks the last iteration in which it was used. If an assortment has not been utilized within

a defined inactivity threshold φ, it is removed from the set of possible assortments. This approach streamlines the optimization process by focusing on active and relevant assortments, reducing computational overhead while maintaining the quality of solutions.

Algorithm 3: Remove Old Assortments Based on Inactivity

Input: Current possible assortments $\mathcal{A}_{\text{poss}}$, last used iteration $\mathcal{I}_{\text{last}}$, current iteration i, inactivity threshold φ.

Output: Updated possible assortments $\mathcal{A}'_{\text{poss}}$.

foreach $(d, q) \in \mathcal{A}_{poss}$ **do**
  foreach *assortment* $\mathcal{S}_q^d \in \mathcal{A}_{poss}(q, d)$ **do**
    Extract last used iteration $\mathcal{I}_{\text{last}}(\mathcal{S}_q^d)$ **if** $i - \mathcal{I}_{last} \geq \varphi$ **then**
      Remove $\mathcal{S}_q^d$ from $\mathcal{A}_{\text{poss}}(q, d)$
return $\mathcal{A}'_{poss} = \mathcal{A}_{poss}$

6. Numerical experiments

This section evaluates the proposed model and solution algorithm through two key experiments: (1) assessing the algorithm's computational performance across different instance settings, and (2) analyzing the impact of commission-based (CB) and fixed employment (FE) compensation policies on operational costs, rider utilization, and profitability, under varying network configurations. Additionally, in Section 6.4, we test the proposed algorithm on Amsterdam restaurant data to find practical trade-offs between rider shift duration and delivery windows, comparing profitability, fleet size and rider workload measures. The computational analysis considers: (i) the effectiveness of algorithm components in reducing computational time, comparing standard Gurobi (`G`), Gurobi with Benders decomposition (`G+B`), and the Iterative Assortment Generation algorithm (`IAG`); and (ii) the profitability and efficiency of the integrated RSRDP with `IAG` versus the separated benchmark model explained in Section 6.1. Results show that the RSRDP consistently outperforms the benchmark in expected profitability while maintaining service quality standards, with `IAG` delivering scalable, practical, and high-quality solutions. Additionally, CB excels in high-variability environments providing high profitability, while FE performs better in stable settings, ensuring balanced fleet utilization.

All computational experiments were conducted on two systems: smaller instances were solved on a virtual machine equipped with an Intel(R) Core(TM) i7-6700HQ CPU, 2.60 GHz processor, and 32 GB of RAM, while larger instances were processed on the DelftBlue supercomputer (Delft High Performance Computing Centre, 2024) with an Intel(R) Xeon(R) Gold 6248R CPU, 3.00 GHz processor, and 185 GB of RAM. All experiments were implemented in `Python 3.9.7` and solved using `Gurobi 11.0.1`.

6.1. Benchmark: separated assortment and rider optimization

We adopt a benchmark inspired by Davis, Gallego, and Topaloglu (2014). The key idea is to separate the assortment selection from the rider decisions, focusing first on which restaurants to offer in each service district and thereafter determining delivery costs based on those selections. We begin by considering a framework in which each service district optimizes its restaurant offerings independently. Under the assumptions of $v_{i0}^d = 0$ and $\gamma_q^d \leq 1 \ \forall d \in D, \ q \in Q$, the set of nested-by-revenue assortments is optimal to consider for the assortment offerings. This means restaurants are ranked according to their expected contribution to revenue, allowing us to include only those that are most profitable. This problem can be solved in polynomial time. Once the best restaurants have been selected for each service district, we calculate the delivery costs by solving the rider problem separately. Because the assortment has already been chosen, this step solely addresses how to dispatch riders and plan their routes to serve the anticipated orders. By combining the revenues from the selected restaurants with the estimated rider costs, we establish an overall expected profit. This outcome serves as our benchmark, reflecting a straightforward practice of treating assortment and routing as separate decisions. It provides the baseline for judging the effectiveness of integrated approaches.

6.2. Computational performance

6.2.1. Instance description. The algorithm was tested on samples of generated datasets using a set of instances. Each instance is defined by the tuple $(D, Q, M, R, v_{q0}^d, [y_{qd}^L, y_{qd}^U])$, where D is the number of service districts, Q the number of cuisine types, M the number of zones, and R the total number of restaurants available. The no-purchase option v_{q0}^d is considered in two configurations: either set to zero, meaning customers always place an order once they choose a cuisine, or assigned a value of 10, where customers have the possibility of opting out, better reflecting real-world behavior where some customers browse without committing to a purchase. The dissimilarity parameters $y_q^d \in [y_{qd}^L, y_{qd}^U])$ are sampled from the ranges $[0, 1]$, indicating competitiveness between within nest restaurants, and $[1, 2]$, indicating synergy between restaurants. These configurations allow us to evaluate different behavioral scenarios and compare against the benchmark where it is assumed that $v_{q0}^d = 0$ and $\gamma_q^d \in [0, 1]$.

The experiments cover a broad range of problem sizes. Smaller instances include $Q = 2$ cuisine types and restaurant counts $R \in \{10, 15, 20\}$, whereas larger instances explore $Q = 4$ with restaurant counts extending to $R \in \{50, 100, 150, 200\}$. The spatial structure remains fixed with $D = 4$ service districts and $M = 15$ zones, ensuring consistency across all test cases. Zones are generated using the `H3` indexing system. Each instance is constructed to reflect real-world conditions, explained in the subsequent section.

6.2.2. Parameter settings. We adopt realistic parameter settings derived from industry information to evaluate the algorithms' performance under practical scenarios. Table 1 summarizes these settings, including spatial and temporal parameters, demand distributions, and cost coefficients. The restaurant and customer distributions are configured to reflect realistic urban settings, with restaurants concentrated 50% in central zones and 50% randomly distributed across peripheral zones.

Table 1 Parameter settings for computational performance test instances.

Parameter	Value	Description
T	48 periods	Total time horizon
D	4 districts	Number of districts
M	15 zones	Number of spatial zones
ρ	$\max(\tau_{mm'})$ periods	Delivery window deadline
η	1 period	Meal preparation time
λ_{mt}	$\sim$ Poisson(2)	Customer demand distribution
$\theta_{\min}$	0 periods	Minimum rider shift duration
$\theta_{\max}$	48 periods	Maximum rider shift duration
$c_{FE}^{overhead}$	€54 euro	Daily overhead costs per rider under FE policy
$c_{CB}^{overhead}$	€18 euro	Daily overhead costs per rider under CB policy
c_{qr}^{d}	€3.75 euro	Delivery cost per order CB policy
c^{t}	€2.50 euro	Time discretized wage for FE riders
φ	5 iterations	Inactivity threshold value `IAG` algorithm
I_{limit}	50 iterations	Iteration limit `IAG` algorithm
T_{limit}	3600 seconds	Computational time limit `IAG` algorithm

Restaurant revenues p_{qr}^{d} and attraction levels v_{qr}^{d} are generated using the methodology of Alfandari, Hassanzadeh, and Ljubić (2021). We sample U_{dqr} from a uniform distribution over $[0, 1]$, and X_{qr}^{d} and Y_{qr}^{d} are independently sampled from a uniform distribution over $[5, 15]$. Then, the revenues and attraction levels are calculated as: $p_{qr}^{d} = 10 \times U_{qr}^{d} \times X_{qr}^{d}$, $v_{qr}^{d} = 10 \times (1 - U_{qr}^{d}) \times Y_{qr}^{d}$ $\forall d \in D,\ q \in Q,\ r \in R$, where higher-priced restaurants tend to have lower attraction levels, aligning with the idea that expensive options appeal to fewer customers. However, random variation ensures that this relationship is not strictly deterministic. The revenue distribution is skewed, producing many low-revenue restaurants and a few high-revenue ones, reflecting real-world restaurant dynamics. Platform revenue is modeled as a percentage (15–30%) of customer orders, consistent with industry standards. To generate realistic profits, customer order values are set based on an average of €34, with revenue and attraction parameters sampled accordingly.

6.2.3. Computational performance results. For all tests, we adopt the FE policy. A time limit of 3600 seconds is imposed on each instance. For instances that do not converge to optimality within this limit, we report the best-found solution and the corresponding optimality gap, as well as its iteration number for the `IAG` algorithm. We solve 20 samples of each instance, and present the average results. To evaluate the relative performance of different models, we calculate the

average percentage improvement for key performance indicators (KPIs), such as computation time, objective value or required fleet size, using the following formula:

$$\text{KPI improvement}(\%) = \frac{1}{20}\sum_{I=1}^{20} 100\% \cdot \left(\frac{|\text{KPI}(1) - \text{KPI}(2)|}{\text{KPI}(1)}\right) \tag{6.1}$$

Here, $I = 1, ..., 20$ denotes each sample, with (1) representing KPI results from Method 1 that is compared to (2), representing results from Method 2.

The effect of different algorithm components. Table 2 presents the objective values, computation times, and optimality gaps for all methods, along with the number of iterations for `IAG`. Table 3 quantifies the performance improvements, showing percentage reductions in time and any changes in objective values. The results highlight the impact of Benders decomposition and the IAG algorithm on solving the RSRDP. While all three methods achieve the same objective values across tested instances, significant differences emerge in computation time. Adding Benders decomposition (`G+B`) substantially reduces computation time compared to using only the standard solver (`G`), with improvements of up to 94.87%. This effect is particularly evident in complex cases with more restaurants, where `G` struggles to close optimality gaps, despite reporting the correct solution. The decomposition effectively strengthens dual bounds, leading to faster convergence.

Further time reductions are observed for `IAG`, outperforming `G+B` by an additional 14% in scenarios with low outside utility values. However, when the outside utility was set higher, `IAG` requires more time than `G+B`, though all instances were still solved within three minutes. Notably, `IAG` scales well as the number of restaurants increases, benefiting from its ability to explore the solution space efficiently without full enumeration. The number of iterations remains low, averaging between 2 and 6, underscoring its rapid convergence. Interestingly, when complexity in terms of available restaurants increases, `IAG` performs relatively better, demonstrating that `IAG` scales well and provides a robust alternative to exact methods for mid-sized instances.

Table 2 The effect of Benders decomposition and IAG.

Instance	`Gurobi`			`Gurobi + Benders`			`IAG`			
	Obj. (€)	Time (s)	Gap (%)	Obj. (€)	Time (s)	Gap (%)	Obj. (€)	Time (s)	Gap (%)	#Iter.
(4,2,15,10,0,[0,1])	39267.59	139.65	0	39267.59	55.86	0	39267.59	47.37	0	2.5
(4,2,15,10,0,[1,2])	73512.25	1551.69	0	73512.25	80.38	0	73512.25	69.03	0	2.1
(4,2,15,10,10,[0,1])	30523.12	112.02	0	30523.12	54.86	0	30523.12	105.25	0	3.9
(4,2,15,10,10,[1,2])	50187.17	1329.27	1.32	50187.17	80.92	0	50187.17	151.46	0	3.9
(4,2,15,15,0,[0,1])	40434.99	1711.60	12.7	40434.99	106.09	0	40434.99	35.81	0	1.6
(4,2,15,15,0,[1,2])	75682.05	3293.44	900.72	75682.51	868.49	0	75682.51	25.15	0	1.5
(4,2,15,15,10,[0,1])	32154.20	1351.24	2.54	32154.20	67.75	0	32154.20	114.76	0	5.8
(4,2,15,15,10,[1,2])	53176.87	2778.13	332.23	53240.35	977.54	0	53240.35	194.16	0	4.5

Table 3 **Performance improvement summary of effects Benders decomposition and IAG algorithm.**

Instance	Obj. % Impr. G → G+B	Obj. % Impr. G+B → IAG	Time % Impr. G → G+B	Time % Impr. G+B → IAG
(4,2,15,10,0,[0,1])	0%	0%	60%	15.20%
(4,2,15,10,0,[1,2])	0%	0%	94.82%	14.12%
(4,2,15,10,10,[0,1])	0%	0%	50.03%	-91.85%
(4,2,15,10,10,[1,2])	0%	0%	93.91%	-87.17%
(4,2,15,15,0,[0,1])	0%	0%	93.80%	66.25%
(4,2,15,15,0,[1,2])	0%	0%	73.63%	97.10%
(4,2,15,15,10,[0,1])	0%	0%	94.87%	-69.39%
(4,2,15,15,10,[1,2])	0.12%	0%	93.01%	80.14%

Comparing the integrated RSRDP to the separated benchmark. The results in Table 4 compare the separated benchmark model, which solves assortment and rider optimization sequentially, with our integrated RSRDP solved with IAG, where both decisions are jointly optimized. The table presents the objective values, the number of riders required, and the percentage improvement in both metrics across different problem instances.

Table 4 **Comparing the separated benchmark to the integrated model.**

Instance	Separated benchmark		Integrated model		% Impr.	
	Obj. value	# Riders	Obj. value	# Riders	Obj. value	# Riders
(4,2,15,10,0,[0,1])	38993.05	27.34	39267.49	23.55	0.70	-13.86
(4,2,15,10,0,[1,2])	72768.28	42.96	73512.25	37.54	1.01	-12.62
(4,2,15,10,10,[0,1])	29612.76	21.96	30523.12	21.71	2.98	-1.14
(4,2,15,10,10,[1,2])	43045.87	25.78	50187.17	28.63	14.23	11.06
(4,2,15,15,0,[0,1])	40399.01	24.87	40435.00	24.57	0.09	-1.21
(4,2,15,15,0,[1,2])	75423.98	43.47	75682.51	40.33	0.34	-7.22
(4,2,15,15,10,[0,1])	31491.59	19.94	32154.20	21.28	2.06	6.72
(4,2,15,15,10,[1,2])	49643.69	26.83	53240.35	33.41	6.76	24.52
(4,2,15,20,0,[0,1])	48635.53	24.35	48730.63	23.91	0.20	-1.81
(4,2,15,20,0,[1,2])	87078.07	40.98	87992.27	39.99	1.04	-2.42
(4,2,15,20,10,[0,1])	37240.32	19.00	39235.44	23.72	5.08	24.84
(4,2,15,20,10,[1,2])	52769.58	25.79	62535.25	34.73	15.6	34.66
(4,4,15,50,0,[0,1])	67906.86	29.39	67980.02	28.73	0.11	-2.25
(4,4,15,50,0,[1,2])	104933.61	41.46	105252.31	39.31	0.30	-5.19
(4,4,15,50,10,[0,1])	48353.21	21.45	52016.95	25.77	7.04	20.14
(4,4,15,50,10,[1,2])	57285.94	24.83	69201.41	30.64	17.22	23.40
(4,4,15,100,0,[0,1])	78849.35	31.34	79029.39	29.90	0.23	-4.59
(4,4,15,100,0,[1,2])	118097.91	41.00	118216.01	38.32	0.10	-6.54
(4,4,15,100,10,[0,1])	59575.97	24.08	64966.45	28.57	8.30	18.65
(4,4,15,100,10,[1,2])	62437.89	20.85	82351.77	31.55	24.18	50.32
(4,4,15,150,0,[0,1])	82233.79	29.47	82302.65	28.69	0.08	-2.65
(4,4,15,150,0,[1,2])	123268.25	39.28	123296.26	38.36	0.02	-2.34
(4,4,15,150,10,[0,1])	62038.43	22.71	67892.55	27.93	8.62	22.99
(4,4,15,150,10,[1,2])	61038.09	20.24	85513.95	31.51	28.62	55.68
(4,4,15,200,0,[0,1])	80393.57	31.15	80525.82	28.99	0.17	-6.93
(4,4,15,200,0,[1,2])	123371.40	43.27	123617.42	40.41	0.20	-6.61
(4,4,15,200,10,[0,1])	62801.29	24.73	68775.88	28.67	8.69	15.93
(4,4,15,200,10,[1,2])	69387.55	21.56	85378.27	33.84	18.73	56.96

The integrated RSRDP model consistently outperforms the separated benchmark in terms of expected profit, with improvements ranging from marginal increases of 0.09% in simpler cases to substantial gains exceeding 28% in more complex settings. The largest improvements occur when the outside utility is high ($v^d_{q0} = 10$) and the restaurants exhibit higher synergy among each other ($\gamma^d_q \in [1, 2]$). This indicates that the benefits of integration grow as users have an option of not purchasing anything when they have already chosen a cuisine type or restaurants are synergistic with respect to each other, or both. Conversely, in cases with no outside utility and high competitiveness, the profit improvements remain modest, often below 1%, the value of joint optimization is less pronounced yet still beneficial. For the, in our opinion, most real-world reflective case of positive outside-utility ($v^d_{q0} = 10$) and competitiveness between restaurants ($\gamma^d_q \in [0, 1]$), the improvements are still significant around 6%. Overall, the impact of integration becomes stronger as the problem size increases, indicating that for a larger real-world network, integration is even more beneficial.

In terms of fleet size, the integrated RSRDP generally reduces the number of required riders, with decreases of up to 13.86%. This efficiency gain results from the coordinated optimization of assortment and delivery, leading to a more compact and effective allocation strategy. However, in some instances, particularly those with high outside utility, the required fleet size increases, sometimes even by over 50%. This is driven by the RSRDP's ability to attract more customers through strategically optimized assortments, leading to a higher order volume that necessitates additional riders. Despite this, the corresponding profit increase is substantial, with instances such as (4,4,15,200,10,[1,2]) showing an 18.73% improvement in profit alongside a 56.96% rise in fleet size. These cases highlight that while more riders are required, the increase is a direct consequence of capturing more market demand and driving higher overall profitability.

Overall, the performance of the integrated approach is strongly influenced by the behavioral characteristics of customers. When customers are more likely to opt out of purchasing, integrating assortment and allocation decisions allows the platform to strategically influence demand, leading to higher revenues and sometimes requiring a larger fleet to meet demand. Similarly, when restaurants are synergistic, customers display stronger preferences for specific options, making optimized assortments significantly more valuable. On the other hand, when outside utility is low and restaurants display higher competitiveness, the impact of integration is limited, yet still beneficial in terms of profitability and reduced fleet size.

6.3. Impact of compensation policies and other managerial insights

6.3.1. Experimental setup. We evaluate commission-based (CB) and fixed employment (FE) compensation policies over a simulated 12-hour operating window (11:00–23:00) in a mid-sized

urban area. The region consists of 34 zones across four service districts, with 100 restaurants spanning 10 cuisine types. Restaurant distribution reflects real-world conditions, with 50% in central zones and 50% in peripheral areas.

Customer arrivals follow a Poisson process with demand peaking during meal times, resulting in 1,250–1,500 daily orders. Assortments assume competitive restaurant interactions ($\gamma_q^d \in [0,1]$), and customers may opt out at the cuisine level ($v_{q0}^d = 10$). Meal preparation takes 10 minutes, and deliveries must be completed within one hour. Rider shifts range from 4 to 12 hours. Under FE, riders earn €15 per hour (€2.50 per 10-minute time step), while under CB, they receive €5 per completed delivery, assuming an average of three deliveries per hour. Hiring costs range from 10–30% of wages, totaling €54 for platform-employed and €18 for independent riders. The RSRDP is solved using `IAG` with a 2-hour time limit and a maximum of 50 iterations. Each scenario is replicated 10 times to ensure robust results.

6.3.2. Influence of network configurations on performance of CB and FE policies. We investigate the impact of several factors on the relative performance of CB and FE, including customer arrival rate fluctuations, restaurant distribution, shift regulations, and delivery deadlines. Summaries of the results can be found in the Tables and Figures presented in Appendix B.

Effect of restaurant distribution. When analyzing three restaurant distributions, base (50% central clustering), centered (90% central clustering), and distributed (even dispersion across all zones), CB consistently outperforms FE in expected profit and revenue by flexibly scaling rider participation and repositioning at low marginal cost. In dispersed areas, CB mobilizes more riders on demand, boosting gains. However, with centralized or uniform restaurant layouts, FE's stable pool of salaried riders can maintain coverage without excessive idle costs, though it lacks CB's dynamic responsiveness.

Impact of delivery window. Short delivery deadlines (20–30 minutes) significantly constrain revenues due to reduced geographic coverage, especially penalizing FE due to its fixed labor costs. Extending deadlines (40–70 minutes) markedly increases revenues and profitability for both CB and FE by expanding service areas and order volumes. CB leverages these longer deadlines more effectively through dynamic rider repositioning and workforce scaling, despite higher delivery costs. FE sees moderate gains but remains hindered by fixed labor expenses. While deadlines beyond 70 minutes may slightly increase profits further, customer satisfaction may deteriorate, emphasizing the optimal range (40–70 minutes) as critical for balancing operational efficiency and customer experience.

Role of shift regulations. Analyzing rider's average shift durations from fully flexible (0h-12h) to strictly fixed (6h), CB yields higher profits through dynamic rider repositioning. Rigid shifts reduce over-staffing and turnover in FE, ensuring predictability. Maximum profitability occurs at moderately flexible shifts (5h-7h) for both CB and FE, optimizing the balance between cost and responsiveness. Thus, platforms prioritizing dynamic scaling benefit from flexibility, while those focused on stability may favor fixed shifts.

Impact of customer arrivals throughout the time horizon. CB consistently outperforms FE across three customer arrival patterns: the base scenario, featuring moderate demand peaks around standard meal times; the uniform scenario, where customer arrivals are steady throughout the day; and the peak scenario, characterized by sharp surges at lunch and dinner. CB leverages dynamic rider repositioning and scalable workforce management to effectively capture revenues, especially during peak demand, despite higher relocation costs. FE, constrained by fixed salaried labor, struggles financially during demand surges but remains relatively competitive under uniform conditions. Thus, CB suits volatile demand patterns, while FE favors predictable, stable demand environments.

Cost considerations in CB. Increasing per-delivery costs under CB (€3-€7) consistently highlights CB's profitability and adaptability compared to FE. Although increasing per-delivery fees raises CB's variable costs, its adaptive deployment keeps marginal expenses low, maintaining higher profit margins than FE. At very high cost levels, the gap narrows as per-trip fees approach the fixed costs of FE. In such cases, a hybrid model, using a salaried core with flexible riders during peaks, may be optimal.

Overall, our findings indicate that there is no one-size-fits-all policy for meal delivery platforms. Commission-based (CB) consistently outperforms fixed employment (FE) in terms of profitability and revenue, particularly in environments characterized by significant demand variability, wide service areas, and extended delivery windows. However, this profitability advantage comes with trade-offs: higher relocation flows under CB can increase rider workload and potentially lead to dissatisfaction. Conversely, FE offers predictable labor costs and stable coverage in scenarios with uniform demand or centralized restaurant distributions, though its fixed cost structure may escalate rapidly under high-demand conditions. Ultimately, the optimal policy depends on balancing profit maximization with operational efficiency and rider satisfaction. Future work might explore hybrid models that integrate the dynamic scalability of CB with the cost stability of FE to further enhance performance across diverse operational scenarios.

6.4. Scaling RSRDP to real-world data

In this section, we evaluate the practical application of the RSRDP by scaling our proposed approach to real-world data from Amsterdam, The Netherlands. Using data from 100 restaurants,

randomly selected to represent a diverse sample spanning 13 cuisines, we analyze how our model performs under realistic urban conditions. The data is obtained from a real meal delivery platform available in Amsterdam and the data includes specifics on the location of the restaurant, main cuisine type, review score (1.0-5.0) and price category (1-3). The study area is defined by the geographical boundaries of Amsterdam, which we partition into five service districts based on existing municipal divisions (Amsterdam 2024a). Each district is covered with the hexagonal zonal structure generated by the `H3` spatial indexing system at a resolution that approximates an 8-minute travel time between adjacent zone centroids.

To simulate customer demand, we combine population density factors (Amsterdam 2024b) with temporal patterns that mirror typical meal-ordering behavior. Specifically, the arrival rate in each zone and time period, λ_{mt}, is computed based on these factors, resulting in an aggregate of roughly 2600 orders over the planning horizon. The attractiveness of restaurants is modeled through a regression that accounts for both price category and review scores. Here, the average price for a restaurant is determined by its price category, with:

$$\text{avg\_price}_r = \begin{cases} 10 & \text{if price category 1} \\ 15 & \text{if price category 2} \\ 30 & \text{if price category 3} \end{cases} \tag{6.2}$$

The restaurant-specific attraction value is based on the negative influence of higher prices, and positively influenced by high review scores, given by:

$$v_{qr}^d = v_{q0}^d + 10 \times (2 \times \text{review}_r - \text{price\_category}_r + \epsilon_{dqr}^{(1)}) \tag{6.3}$$

Where review_r denotes the review score of restaurant r, and $\epsilon_{dqr}^{(2)} \sim$ extreme value type I $=$ Gumbel($\mu = 0, \beta = 1$) accounts for unobserved factors. Revenues from restaurants are based on the price category, using the commission rate of 15%-30% per order. We draw commission rates for restaurant r from the uniform distribution: $\text{commission\_rate}_r \sim U(0.15, 0.30)$. The price parameter is then calculated as:

$$p_{qr}^d = 2 \times \text{avg\_price}_r \times \text{commission\_rate}_r \times (1 + \epsilon_{qrd}^{(2)}) \tag{6.4}$$

Where the multiplicative noise term is $\epsilon_{qrd}^{(2)} \sim N(0, 0.05)$. This formulation means that the base price is scaled by 2 and then adjusted for the commission and a small normally distributed perturbation. Other parameters and sets are consistent with those described in Subsection 6.3, and Figure 6 provides an overview of the service districts, zonal structure, demand distribution, and restaurant locations, for the Amsterdam case study.

The selected case study reflects the typical urban distribution of restaurants, with a high concentration in the city center. Through this analysis, we investigate the impact of varying average shift durations and maximum delivery windows on key performance metrics, including platform profitability, rider workload per hour, and the required fleet size. Our objective is to identify a Pareto-optimal trade-off that can inform decision-making for meal delivery platforms. The RSRDP is solved under the Fixed Employment (FE) policy to evaluate its implications in this setting.

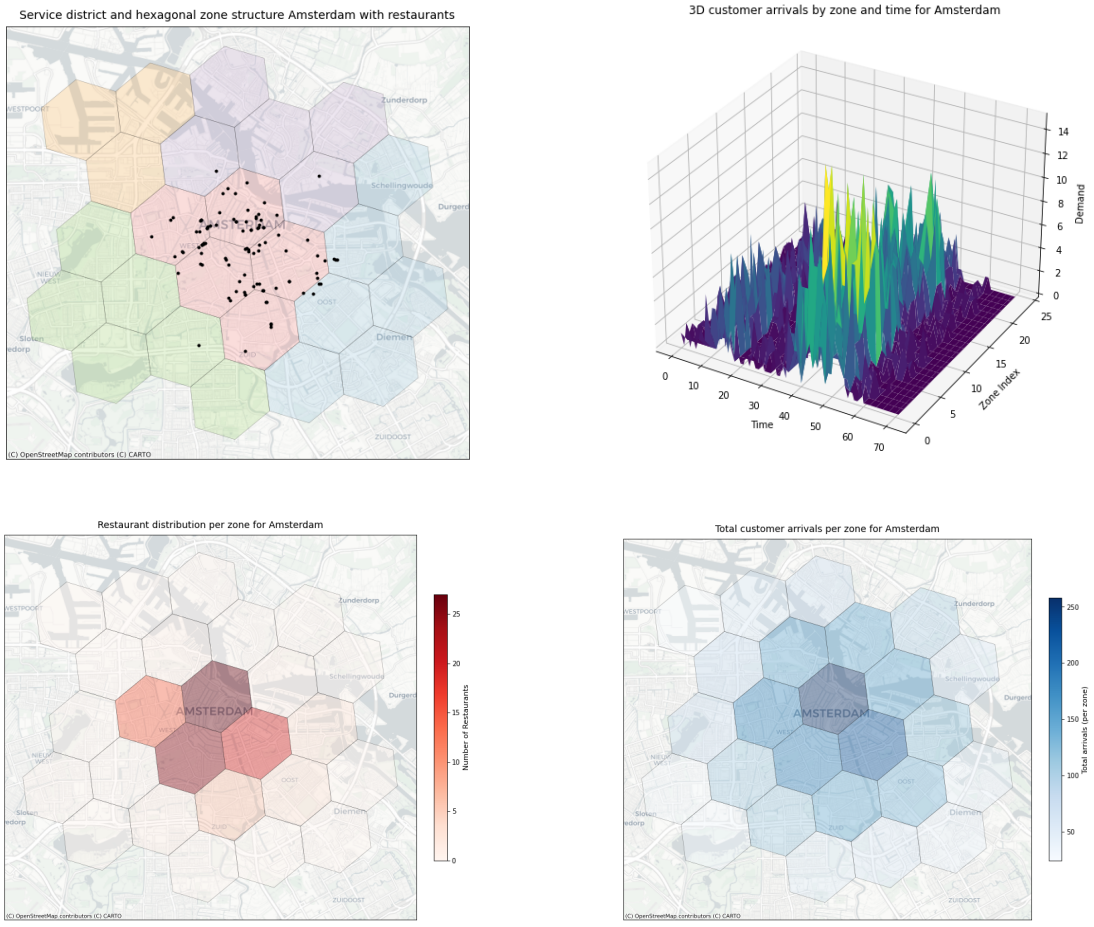

Figure 6 **Case study data of Amsterdam. Top left indicates service regions, zones and restaurant locations. Top right presents customer arrivals over space and time for the time horizon. Bottom left shows heatmap of restaurant locations, bottom right heatmap of total customer arrivals in space over time horizon.**

We define each solution as a vector $(Profit, \#R, WL)$, where we want to maximize the $Profit$, minimize the number of required riders $\#R$, and either minimize or have a reasonable workload WL for the riders, presented as the average number of orders per hour per rider. Mathematically,

we say that a solution $(Profit_i, \#R_i, WL_i)$ dominates another solution $(Profit_j, \#R_j, WL_j)$ if the following conditions hold:

$$Profit_i \geq Profit_j \qquad \#R_i \leq \#R_j \qquad WL_i \leq WL_j \tag{6.5}$$

With at least one of these equalities being strict. In other words, $(Profit_i, \#R_i, WL_i)$ is considered better than $(Profit_j, \#R_j, WL_j)$ in all objectives, without being worse in any. The Pareto frontier consists of all solutions that are not dominated by any other solution in the set.

Table 5 Case study results.

		Delivery window [minutes]														
		30			40			50			60			70		
		Profit	# R	WL	Profit	# R	WL	Profit	# R	WL	Profit	# R	WL	Profit	# R	WL
Average shift duration [hours]	[4-6]	13263.07	87.23	3.64	15471.96	96.30	3.55	20274.53	76.79	4.66	23443.93	57.96	6.95	24045.89	57.36	7.20
	[5-7]	13140.89	86.52	3.14	14884.17	91.98	3.10	19177.51	76.68	3.89	22446.59	59.42	5.28	22761.63	46.85	6.37
	[6-8]	11801.41	85.85	2.70	16265.18	62.13	3.85	18044.48	68.08	3.69	21634.53	53.53	4.91	23029.38	47.96	5.69
	[7-9]	10691.79	64.68	2.65	13371.34	53.14	3.43	18805.00	64.36	3.56	21421.72	56.59	4.26	22315.70	46.85	4.77
	[8-10]	9583.14	64.16	2.34	12956.40	72.42	2.57	16942.20	70.62	2.90	20851.20	53.79	3.99	21659.90	44.83	4.43

The experimental evaluation examines the effects of varying two key operational parameters: the average rider shift duration and the maximum delivery window. Table 5 summarizes the outcomes in terms of platform profitability, the required number of riders (denoted as #R), and the average hourly workload, i.e. number of deliveries, per rider (WL). For instance, with a shift duration of 4–6 hours, increasing the delivery window from 30 to 70 minutes results in a profit increase from \$13,263.07 to \$24,045.89, yet the rider workload also rises from 3.64 to 7.20 deliveries per hour. In contrast, longer shift durations tend to yield lower overall profitability but generally correspond to lower workloads, highlighting a trade-off between rider efficiency and cost-effectiveness.

The analysis is further enriched by the identification of Pareto frontier solutions. These solutions reveal balanced configurations where incremental increases in profitability are accompanied by relative changes in fleet size and rider workload. Figure 7 presents contrasting facets of the Pareto frontier solutions in a different context. The left subplot showcases the Pareto frontier considering profit, number of riders, and workload, where each point is color-coded to reflect workload intensity. For example, one Pareto optimal solution with a 4–6 hour shift and a 60-minute delivery window yields a profit of \$23,443.93 with 57.96 riders and an average workload of 6.95 deliveries per hour, while another solution with a 5–7 hour shift and a 50-minute window achieves \$19,177.51 profit with 76.68 riders and a workload of 3.89 deliveries per hour. Such comparisons underscore that a moderate delivery window, paired with an appropriate shift duration, can enhance service quality by keeping rider workloads within a reasonable threshold (around 3 deliveries per hour) while simultaneously improving profitability. The right subplot narrows the focus to the Pareto

optimal solutions based solely on profit and the number of riders, while still employing workload for visual clarity. This simplification underscores the pivotal role of workload management in balancing service quality and operational efficiency. Notably, solutions situated in the bottom-right quadrant of both subplots exemplify the platform's capacity to maximize profitability with fewer riders, albeit with varying workload implications and a large delivery window.

These findings provide actionable insights for meal delivery platforms. In practice, shorter delivery windows enhance customer satisfaction and service quality, but may necessitate either more riders and lower profits. Conversely, extending the delivery window improves profit margins but risks overburdening riders and compromising timely service. For decision makers, the Pareto frontier serves as a decision-support tool, enabling a trade-off analysis where relative changes can be carefully evaluated.

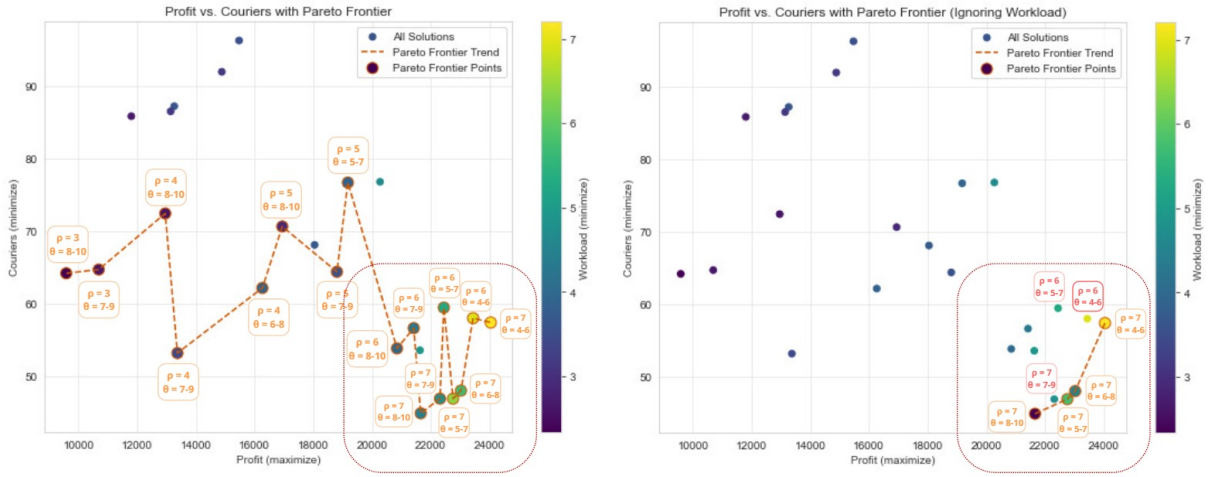

Figure 7 **The 2D Pareto frontier solutions display profit on the x-axis and the number of riders on the y-axis, with color representing workload. The left figure considers dominance across all three metrics, profit, number of riders, and workload, while the right figure accounts only for profit and number of riders.**

7. Conclusion

This study introduced the Restaurant Selection and Rider Dimensioning Problem (RSRDP) that jointly considers decisions on restaurant assortments and rider dimensioning to enhance profitability and service quality for meal delivery platforms. Numerical experiments confirm that designing meal delivery platform services using our proposed model outperforms conventional approaches where these steps are optimized in sequence. The proposed iterative assortment generation algorithm demonstrates practical scalability.

Further, by evaluating two rider compensation schemes, commission-based (CB) and fixed employment (FE), we observe that CB leverages flexible deployment to sustain higher profit margins, especially when demand is volatile. Meanwhile, EC models tend to yield lower average profitability in volatile contexts, but offer the advantages of workforce stability, cost predictability, and potentially smaller fleets. These insights suggest that real-world operators may benefit from hybrid compensation strategies, blending a core set of employed riders with a pool of on-demand freelancers to handle peak loads while stabilizing baseline services.

Moving forward, future research could expand this framework in several directions. One promising avenue is exploring more complex hybrid compensation schemes that dynamically adjust the mix of employed and freelance couriers based on the time of day, demand forecasts, or regional characteristics. Additionally, incorporating multi-objective criteria, such as balancing profit with customer satisfaction or driver well-being, would provide a more holistic view of performance. Time-varying assortments, which adjust to changing consumer preferences throughout the day, could refine demand modeling. Finally, developing more advanced heuristic and decomposition approaches would further reduce computational complexity, especially when tracking individual couriers. These extensions can equip meal delivery platforms with even more robust, profitable, and equitable strategies to serve consumers' growing expectations.

Acknowledgements

This research was undertaken as part of the project SINERGI, funded by JPI-Urban Europe and NSFC.

References

Alfandari L, Hassanzadeh A, Ljubić I, 2021 *An exact method for assortment optimization under the nested logit model. European Journal of Operational Research* 291(3):830–845, URL `http://dx.doi.org/10.1016/j.ejor.2020.12.007`.

Amsterdam G, 2024a *Amsterdam districts and neighborhoods.* URL `https://www.amsterdamsights.com/about/neighborhoods.html`.

Amsterdam G, 2024b *Population Density Amsterdam — Website Onderzoek en Statistiek.* URL `https://onderzoek.amsterdam.nl/interactief/dashboard-kerncijfers?tab=indicator&thema=bevolking&indicator=BEVDICHT&indeling=wijken&jaar=2024&gebied=KA&taal=en`.

Aparicio D, Prelec D, Zhu W, 2025 *Choice Overload and the Long Tail: Consideration Sets and Purchases in Online Platforms. Manufacturing & Service Operations Management* msom.2021.0318, URL `http://dx.doi.org/10.1287/msom.2021.0318`.

Bell MGH, Le DT, Bhattacharjya J, Geers G, 2024 *On-Demand Meal Delivery: A Markov Model for Circulating Couriers. Transportation Science* trsc.2024.0513, URL http://dx.doi.org/10.1287/trsc.2024.0513.

Carlsson JG, Liu S, Salari N, Yu H, 2021 *Provably Good Region Partitioning for On-Time Last-Mile Delivery. SSRN Electronic Journal* URL http://dx.doi.org/10.2139/ssrn.3915544.

Charnes A, Cooper WW, 1973 *An explicit general solution in linear fractional programming. Naval Research Logistics Quarterly* 20(3):449–467, URL http://dx.doi.org/10.1002/nav.3800200308.

Chua BL, Karim S, Lee S, Han H, 2020 *Customer Restaurant Choice: An Empirical Analysis of Restaurant Types and Eating-Out Occasions. International Journal of Environmental Research and Public Health* 17(17):6276, URL http://dx.doi.org/10.3390/ijerph17176276.

Davis JM, Gallego G, Topaloglu H, 2014 *Assortment Optimization Under Variants of the Nested Logit Model. Operations Research* 62(2):250–273, URL http://dx.doi.org/10.1287/opre.2014.1256.

Delft High Performance Computing Centre,, 2024 *DelftBlue Supercomputer (Phase 2)*. https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2.

Fakfare P, 2021 *Influence of service attributes of food delivery application on customers' satisfaction and their behavioural responses: The IPMA approach. International Journal of Gastronomy and Food Science* 25:100392, URL http://dx.doi.org/10.1016/j.ijgfs.2021.100392.

Kancharla SR, Van Woensel T, Waller ST, Ukkusuri SV, 2024 *Meal Delivery Routing Problem with Stochastic Meal Preparation Times and Customer Locations. Networks and Spatial Economics* URL http://dx.doi.org/10.1007/s11067-024-09643-1.

Ke J, Wang C, Li X, 2022 *Equilibrium Analysis for On-Demand Food Delivery Markets. SSRN Electronic Journal* URL http://dx.doi.org/10.2139/ssrn.4291481.

Li J, Yang S, Pan W, Xu Z, Wei B, 2022 *Meal delivery routing optimization with order allocation strategy based on transfer stations for instant logistics services. IET Intelligent Transport Systems* 16(8):1108–1126, URL http://dx.doi.org/10.1049/itr2.12206.

Li X, Ke J, Yang H, Wang H, Zhou Y, 2024 *An aggregate matching and pick-up model for mobility-on-demand services. Transportation Research Part B: Methodological* 190:103070, URL http://dx.doi.org/10.1016/j.trb.2024.103070.

Li Z, Wang G, 2024 *On-Demand Delivery Platforms and Restaurant Sales. Management Science* mnsc.2021.01010, URL http://dx.doi.org/10.1287/mnsc.2021.01010.

Liu S, He L, Shen ZJM, 2018 *Data-Driven Order Assignment for Last Mile Delivery. SSRN Electronic Journal* URL http://dx.doi.org/10.2139/ssrn.3179994.

Liu S, Luo Z, 2023 *On-Demand Delivery from Stores: Dynamic Dispatching and Routing with Random Demand. Manufacturing & Service Operations Management* 25(2):595–612, URL http://dx.doi.org/10.1287/msom.2022.1171.

Steever Z, Karwan M, Murray C, 2019 *Dynamic courier routing for a food delivery service. Computers & Operations Research* 107:173–188, URL http://dx.doi.org/10.1016/j.cor.2019.03.008.

Tang CS, Bai J, So KC, Chen XM, Wang H, 2016 *Coordinating Supply and Demand on an On-Demand Platform: Price, Wage, and Payout Ratio. SSRN Electronic Journal* URL http://dx.doi.org/10.2139/ssrn.2831794.

Ulmer MW, Savelsbergh M, 2020 *Workforce Scheduling in the Era of Crowdsourced Delivery. Transportation Science* 54(4):1113–1133, URL http://dx.doi.org/10.1287/trsc.2020.0977.

Ulmer MW, Thomas BW, Campbell AM, Woyak N, 2021 *The Restaurant Meal Delivery Problem: Dynamic Pickup and Delivery with Deadlines and Random Ready Times. Transportation Science* 55(1):75–100, URL http://dx.doi.org/10.1287/trsc.2020.1000.

Xue G, Wang Z, Wang G, 2021 *Optimization of Rider Scheduling for a Food Delivery Service in O2O Business. Journal of Advanced Transportation* 2021:1–15, URL http://dx.doi.org/10.1155/2021/5515909.

Yang Y, Umboh SW, Ramezani M, 2024 *Freelance drivers with a decline choice: Dispatch menus in on-demand mobility services for assortment optimization. Transportation Research Part B: Methodological* 190:103082, URL http://dx.doi.org/10.1016/j.trb.2024.103082.

Yildiz B, Savelsbergh M, 2019a *Provably High-Quality Solutions for the Meal Delivery Routing Problem. Transportation Science* 53(5):1372–1388, URL http://dx.doi.org/10.1287/trsc.2018.0887.

Yildiz B, Savelsbergh M, 2019b *Service and capacity planning in crowd-sourced delivery. Transportation Research Part C: Emerging Technologies* 100:177–199, URL http://dx.doi.org/10.1016/j.trc.2019.01.021.

Appendix A: Table of notations

Sets and indices		
D	set of service districts	$d \in D$
R	set of restaurants	$r \in R$
Q	set of cuisine types	$q \in Q$
R_m	set of restaurants within delivery limit of zone m	$r \in R_m$
$\hat{R}_m$	set of restaurants located in zone m	$r \in \hat{R}_m$
R_d	set of restaurants within delivery limit for service district d	$r \in R_d$
R_q^d	set of restaurants within delivery limit for cuisine q and service district d	$r \in R_q^d$
M	set of zones	$m, m' \in M$
T	set of time periods such that $T = \{0, \kappa, 2\kappa, ..., T_{max}\}$	$t, t' \in T$
N	set of spatial-temporial nodes (m,t) such that $m \in M$, $t \in T$	$(m,t) \in N$
A	set of arcs	$a \in A$
$A^+_{(m,t)}$	set of possible destination arcs from node (m,t)	$(m',t') \in A^+_{(m,t)}$
$A^-_{(m,t)}$	set of possible origin arcs to node (m,t)	$(m',t') \in A^-_{(m,t)}$
Parameters		
$\tau_{mm'}$	travel time from zone m to m'; interzonal travel time $\tau_{mm} = \kappa$	[periods]
ρ	delivery time deadline	[periods]
η	meal preparation time	[periods]
λ_{mt}	number of customer arrivals in zone m and period t	[orders]
$c^{overhead}_{policy}$	overhead cost per required rider for business policies	[euro]
c_a	courier salary on arc a	[euro]
c^t	discretized salary per period t	[euro]
c^d_{qr}	cost per service of delivery ordered from restaurant r of cuisine q in district d	[euro]
p^d_{qr}	expected revenue per order at restaurant r for cuisine type q for service district d	[euro]
v^d_{qr}	attraction value of restaurant r for cuisine type q for service district d	[attraction]
v^d_0	attraction value no-purchase option cuisine level for service district d	[attraction]
v^d_{q0}	attraction value no-purchase option restaurant level for cuisine q and district d	[attraction]
γ^d_q	dissimilarity parameter for cuisine type q for service district d	[dissimilarity]
θ_{min}	average minimum shift duration	[periods]
θ_{max}	average maximum shift duration	[periods]
e^r_q	binary parameter indicating if restaurant r is of cuisine q	[binary]
b^d_m	binary parameter indicating if zone m is covered by service district d	[binary]
Variables		
$z_{S^d_q}$	if assortment S^d_q offered for cuisine nest q for service district d	[binary]
w_a	courier flow on arc a	[continuous]
$u^{in}_{(m,t)}$	number of couriers starting their work at period t in zone m	[continuous]
$u^{out}_{(m,t)}$	number of couriers leaving the system at period t in zone m	[continuous]
y_d; f_d; g_d	auxiliary variables for model formulation	[continuous]
x_{md}; h_{md}	auxiliary variables for model formulation	[continuous]
$l_{S^d_q}$; $k^m_{S^d_q}$	auxiliary variables for model formulation	[continuous]

Appendix B: Tables of impact compensation policies under different network configurations

Table 7 Summary of average results for different restaurant distributions and policies.

Distribution	Policy	Exp. profit	Revenue	Deliv. cost	Hiring cost	# Riders	% Reloc. flow	% Open rest.
Base	FE	52285.31	58107.62	4478.80	2018.52	37.38	16.86	49.75
	CB	54510.21	60843.35	5616.92	716.22	39.79	18.00	51.50
Distributed	FE	49034.86	54971.13	3258.93	2003.38	37.10	26.07	48.60
	CB	53195.40	59676.61	5622.69	858.52	47.70	28.57	57.80
Centered	FE	52945.24	58684.89	3636.57	2103.09	38.95	14.82	55.00
	CB	56335.96	62878.26	5784.42	757.88	42.10	19.85	57.20

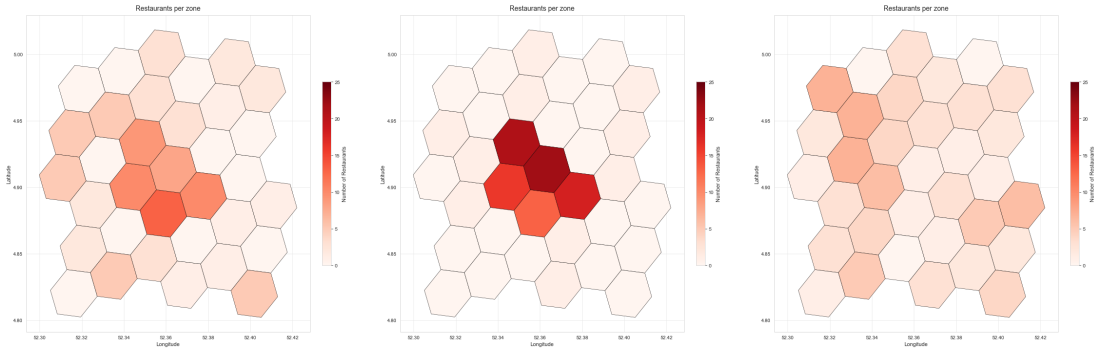

Figure 8 Heatmaps representing examples of restaurant distribution over operating area, base case (left), centered (middle) and distributed (right).

Table 8 Summary of average results for different shift durations and policies.

Shift duration	Policy	Exp. profit	Revenue	Deliv. cost	Hiring cost	# Riders	% Reloc. flow	% Open rest.
[0,12]	FE	55553.44	60601.25	3228.04	1819.77	33.70	0.00	50.75
	CB	56370.05	62520.28	5532.34	617.89	34.33	7.26	54.50
[1,11]	FE	49216.07	54862.81	3555.45	2091.28	38.73	0.13	46.75
	CB	54093.07	60428.1	5588.10	746.94	41.50	12.41	53.00
[2,10]	FE	50234.02	55979.93	3604.18	2141.73	39.66	1.03	49.40
	CB	51932.77	58177.77	5581.64	663.36	36.85	12.06	50.80
[3,9]	FE	50661.00	55826.02	3218.91	1946.12	36.04	2.81	48.25
	CB	53639.53	59932.51	5582.71	710.27	39.46	13.99	54.50
[4,8]	FE	47503.50	53255.96	3761.50	1990.96	36.87	19.06	47.60
	CB	54374.18	61030.91	5840.59	816.15	45.34	20.81	55.20
[5,7]	FE	53443.48	59690.76	4276.31	1970.97	36.50	35.97	52.00
	CB	57115.63	63460.97	5541.42	803.92	44.66	41.41	57.25
[6,6]	FE	46954.00	52430.81	3886.01	1590.80	29.46	44.74	42.80
	CB	49357.30	55214.75	5237.34	620.11	34.45	47.31	49.80

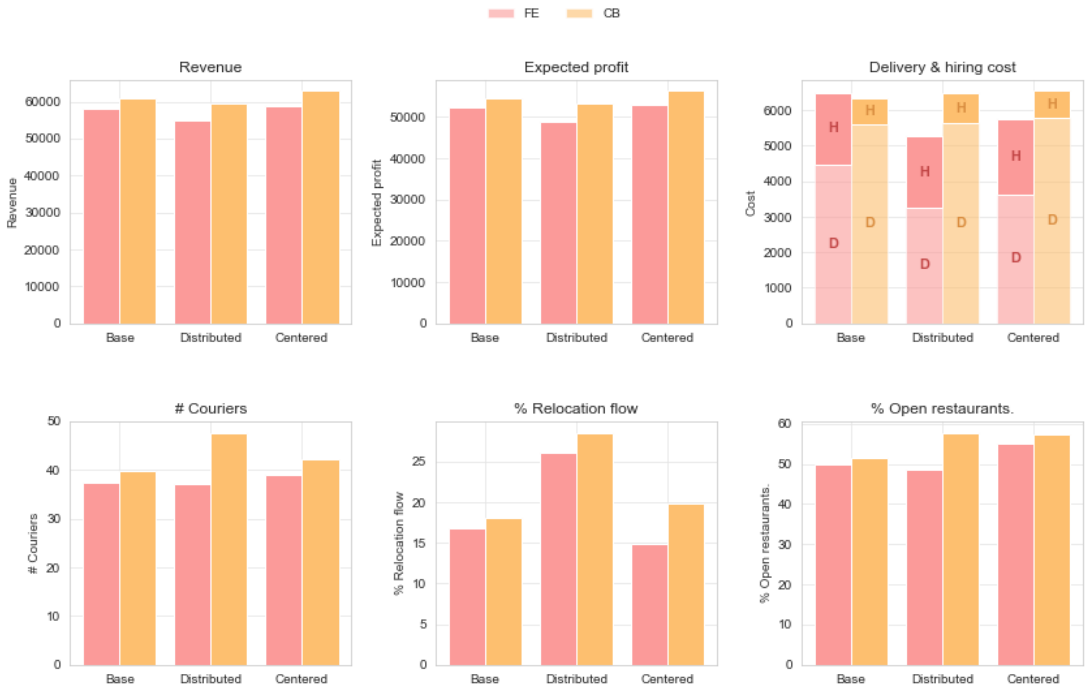

Figure 9 **Plots of performance of the FE and CB policy on different metrics for evaluating the impact of restaurant distributions.**

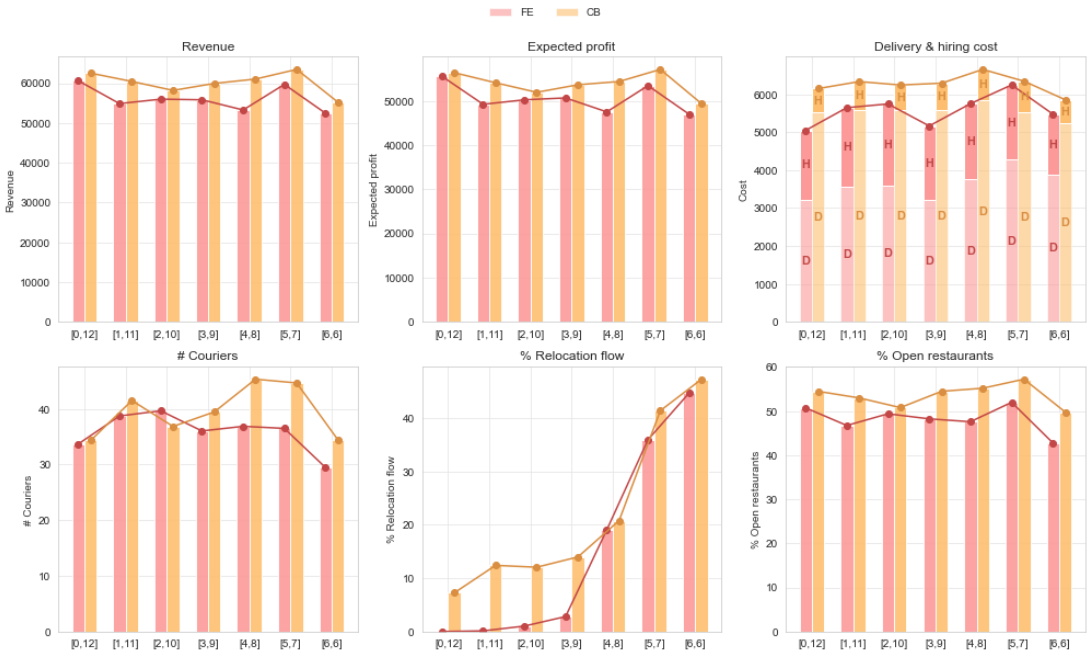

Figure 10 **Plots of performance of the FE and CB policy on different metrics for evaluating the impact of shift durations.**

Table 9 **Summary of average results for different maximum delivery deadlines.**

Max. delivery time [min]	Policy	Exp. profit	Revenue	Deliv. cost	Hiring cost	# Riders	% Reloc. flow	% Open rest.
20	FE	2884.92	4018.15	619.06	514.17	9.52	60.40	4.40
	CB	3404.78	3713.64	154.93	153.92	8.55	78.42	4.40
30	FE	18162.72	22847.46	2731.12	1953.67	36.18	50.47	26.00
	CB	20280.12	22647.23	1627.91	739.20	41.07	51.91	27.20
40	FE	40416.87	47520.43	4291.28	2812.28	52.08	45.17	42.25
	CB	46424.02	53165.75	5615.27	1126.46	62.58	49.19	51.50
50	FE	45058.67	52951.65	4943.67	2949.31	54.62	40.62	47.40
	CB	50210.90	56939.39	5670.34	1058.15	58.79	39.74	54.80
60	FE	43203.56	49447.56	3979.51	2264.49	41.94	27.81	50.75
	CB	48829.63	55185.78	5464.12	892.04	49.56	33.48	57.50
70	FE	49896.55	55646.27	3166.23	1993.63	36.92	21.10	49.20
	CB	54750.86	61275.00	5763.18	760.97	42.27	20.41	56.20
80	FE	51008.30	55907.55	3288.10	1611.15	29.84	9.33	47.50
	CB	54518.39	60620.12	5467.45	634.29	35.24	11.04	48.75

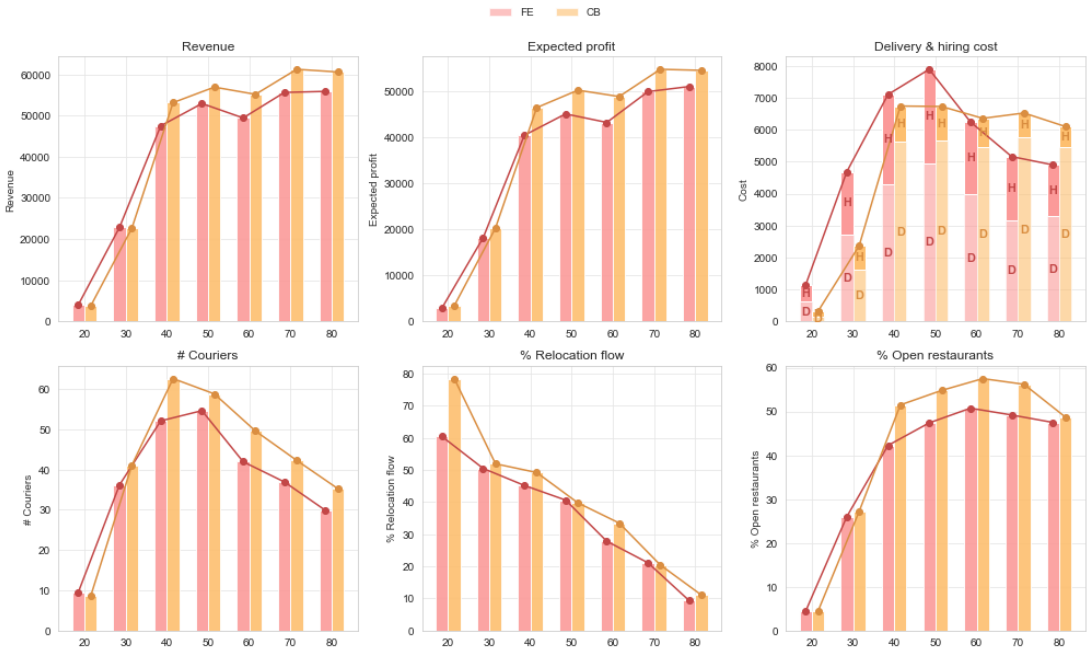

Figure 11 **Plots of performance of the FE and CB policy on different metrics for evaluating the impact of maximum delivery deadlines.**

Table 10 **Summary of average results for different customer arrival distributions.**

Shift duration	Policy	Exp. profit	Revenue	Deliv. cost	Hiring cost	# Riders	% Reloc. flow	% Open rest.
Base	FE	47499.62	53669.36	3957.04	2212.71	40.97	30.94	48.80
	CB	52468.50	58807.99	5485.07	854.42	47.47	35.92	53.60
Uniform	FE	52076.55	56720.65	3501.11	1143.00	21.17	0	50.75
	CB	54999.36	61205.55	5764.76	441.44	24.52	1.84	55.25
Centered	FE	50711.98	58090.73	4585.89	2792.87	51.72	41.16	52.4
	CB	54215.03	60956.75	5594.48	1147.24	63.73	47.26	56.6

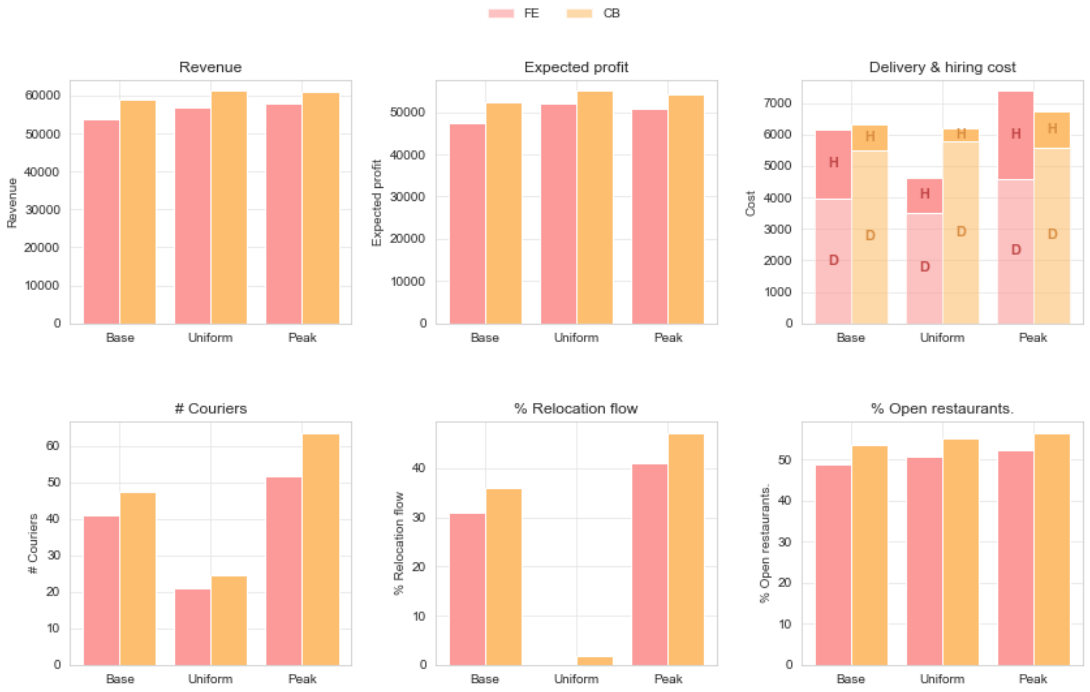

Figure 12 **Plots of performance of the FE and CB policy on different metrics for evaluating the impact of various customer arrival distributions.**

Table 11 **Summary of average results for different delivery costs c^d_{qr}.**

Cost c^d_{qr}	Policy	Exp. profit	Revenue	Deliv. cost	Hiring cost	# Riders	% Reloc. flow	% Open rest.
c = 3	FE	47817.17	55118.73	4710.76	2590.80	47.98	38.21	50.60
	CB	53408.95	57663.16	3353.76	900.45	50.02	38.05	51.20
c = 3.5	FE	46578.77	53069.63	4156.23	2334.62	43.23	29.61	49.40
	CB	50699.13	55343.49	3794.48	849.89	47.22	34.04	52.20
c = 4	FE	45824.30	51841.77	3856.30	2161.17	40.02	32.62	44.60
	CB	52015.58	57451.08	4539.07	896.44	49.80	33.91	54.60
c = 4.5	FE	46871.61	53200.12	4079.78	2248.72	41.64	34.27	46.80
	CB	54482.87	60423.81	5031.00	909.94	50.55	33.91	56.50
c = 5 (base)	FE	50049.26	56285.42	3997.19	2238.97	41.46	29.74	47.40
	CB	53858.38	60647.64	5866.41	922.86	51.27	31.45	54.20
c = 5.5	FE	50904.79	57206.45	4047.54	2254.13	41.74	26.13	49.60
	CB	53394.51	60630.25	6294.56	941.18	52.29	35.57	54.00
c = 6	FE	44754.77	50832.16	3896.01	2181.38	40.39	35.41	45.80
	CB	49421.64	57066.62	6734.86	910.12	50.56	35.20	52.20
c = 6.5	FE	49405.70	55879.90	4156.19	2318.01	42.93	30.70	48.80
	CB	52922.34	61383.76	7489.24	972.18	54.01	35.65	57.20
c = 7	FE	48595.06	55259.37	4279.24	2385.07	44.17	34.81	47.60
	CB	51183.03	60079.34	8025.81	870.50	48.36	33.17	51.00

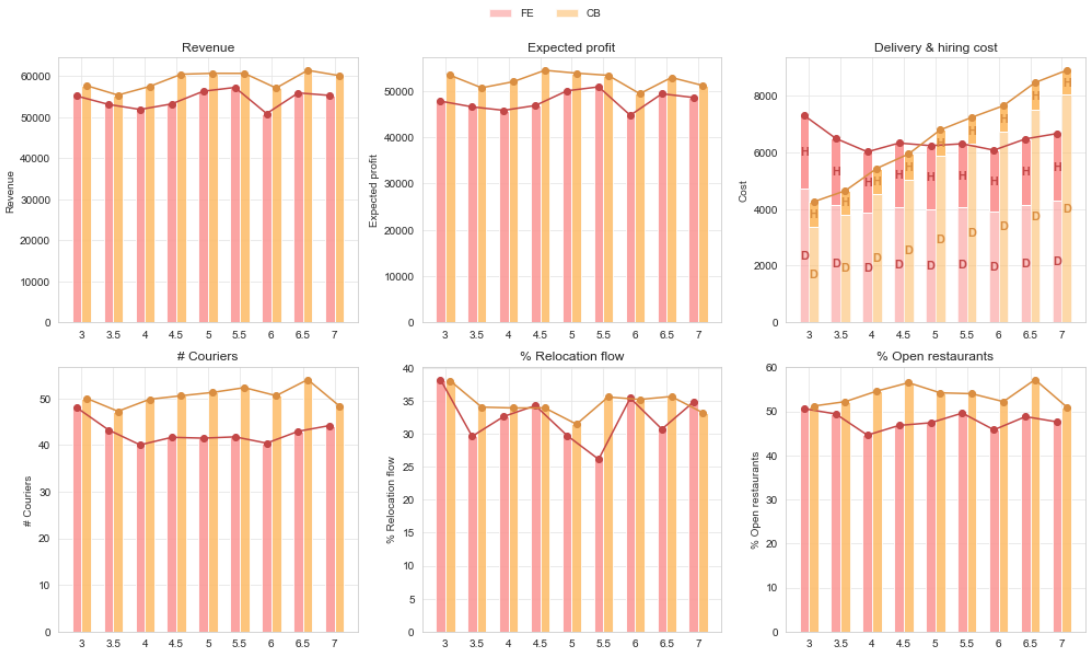

Figure 13 **Plots of performance of the FE and CB policy on different metrics for evaluating the impact of changing c_{qr}^{d}.**

B

Summary tables results compensation policies and other managerial insights

Table B.1: Summary of average results for different restaurant distributions and policies.

Distribution	Policy	Exp. profit	Revenue	Deliv. cost	Hiring cost	# Riders	% Reloc. flow	% Open rest.
Base	FE	52285.31	58107.62	4478.80	2018.52	37.38	16.86	49.75
	CB	54510.21	60843.35	5616.92	716.22	39.79	18.00	51.50
Distributed	FE	49034.86	54971.13	3258.93	2003.38	37.10	26.07	48.60
	CB	53195.40	59676.61	5622.69	858.52	47.70	28.57	57.80
Centered	FE	52945.24	58684.89	3636.57	2103.09	38.95	14.82	55.00
	CB	56335.96	62878.26	5784.42	757.88	42.10	19.85	57.20

Table B.2: Summary of average results for different shift durations and policies.

Shift duration	Policy	Exp. profit	Revenue	Deliv. cost	Hiring cost	# Riders	% Reloc. flow	% Open rest.
[0,12]	FE	55553.44	60601.25	3228.04	1819.77	33.70	0.00	50.75
	CB	56370.05	62520.28	5532.34	617.89	34.33	7.26	54.50
[1,11]	FE	49216.07	54862.81	3555.45	2091.28	38.73	0.13	46.75
	CB	54093.07	60428.1	5588.10	746.94	41.50	12.41	53.00
[2,10]	FE	50234.02	55979.93	3604.18	2141.73	39.66	1.03	49.40
	CB	51932.77	58177.77	5581.64	663.36	36.85	12.06	50.80
[3,9]	FE	50661.00	55826.02	3218.91	1946.12	36.04	2.81	48.25
	CB	53639.53	59932.51	5582.71	710.27	39.46	13.99	54.50
[4,8]	FE	47503.50	53255.96	3761.50	1990.96	36.87	19.06	47.60
	CB	54374.18	61030.91	5840.59	816.15	45.34	20.81	55.20
[5,7]	FE	53443.48	59690.76	4276.31	1970.97	36.50	35.97	52.00
	CB	57115.63	63460.97	5541.42	803.92	44.66	41.41	57.25
[6,6]	FE	46954.00	52430.81	3886.01	1590.80	29.46	44.74	42.80
	CB	49357.30	55214.75	5237.34	620.11	34.45	47.31	49.80

Table B.3: Summary of average results for different maximum delivery deadlines.

Max. delivery time [min]	Policy	Exp. profit	Revenue	Deliv. cost	Hiring cost	# Riders	% Reloc. flow	% Open rest.
20	FE	2884.92	4018.15	619.06	514.17	9.52	60.40	4.40
	CB	3404.78	3713.64	154.93	153.92	8.55	78.42	4.40
30	FE	18162.72	22847.46	2731.12	1953.67	36.18	50.47	26.00
	CB	20280.12	22647.23	1627.91	739.20	41.07	51.91	27.20
40	FE	40416.87	47520.43	4291.28	2812.28	52.08	45.17	42.25
	CB	46424.02	53165.75	5615.27	1126.46	62.58	49.19	51.50
50	FE	45058.67	52951.65	4943.67	2949.31	54.62	40.62	47.40
	CB	50210.90	56939.39	5670.34	1058.15	58.79	39.74	54.80
60	FE	43203.56	49447.56	3979.51	2264.49	41.94	27.81	50.75
	CB	48829.63	55185.78	5464.12	892.04	49.56	33.48	57.50
70	FE	49896.55	55646.27	3166.23	1993.63	36.92	21.10	49.20
	CB	54750.86	61275.00	5763.18	760.97	42.27	20.41	56.20
80	FE	51008.30	55907.55	3288.10	1611.15	29.84	9.33	47.50
	CB	54518.39	60620.12	5467.45	634.29	35.24	11.04	48.75

Table B.4: Summary of average results for different customer arrival distributions.

Shift duration	Policy	Exp. profit	Revenue	Deliv. cost	Hiring cost	# Riders	% Reloc. flow	% Open rest.
Base	FE	47499.62	53669.36	3957.04	2212.71	40.97	30.94	48.80
	CB	52468.50	58807.99	5485.07	854.42	47.47	35.92	53.60
Uniform	FE	52076.55	56720.65	3501.11	1143.00	21.17	0	50.75
	CB	54999.36	61205.55	5764.76	441.44	24.52	1.84	55.25
Centered	FE	50711.98	58090.73	4585.89	2792.87	51.72	41.16	52.4
	CB	54215.03	60956.75	5594.48	1147.24	63.73	47.26	56.6

Table B.5: Summary of average results for different delivery costs c^d_{qr}.

Cost c^d_{qr}	Policy	Exp. profit	Revenue	Deliv. cost	Hiring cost	# Riders	% Reloc. flow	% Open rest.
c = 3	FE	47817.17	55118.73	4710.76	2590.80	47.98	38.21	50.60
	CB	53408.95	57663.16	3353.76	900.45	50.02	38.05	51.20
c = 3.5	FE	46578.77	53069.63	4156.23	2334.62	43.23	29.61	49.40
	CB	50699.13	55343.49	3794.48	849.89	47.22	34.04	52.20
c = 4	FE	45824.30	51841.77	3856.30	2161.17	40.02	32.62	44.60
	CB	52015.58	57451.08	4539.07	896.44	49.80	33.91	54.60
c = 4.5	FE	46871.61	53200.12	4079.78	2248.72	41.64	34.27	46.80
	CB	54482.87	60423.81	5031.00	909.94	50.55	33.91	56.50
c = 5 (base)	FE	50049.26	56285.42	3997.19	2238.97	41.46	29.74	47.40
	CB	53858.38	60647.64	5866.41	922.86	51.27	31.45	54.20
c = 5.5	FE	50904.79	57206.45	4047.54	2254.13	41.74	26.13	49.60
	CB	53394.51	60630.25	6294.56	941.18	52.29	35.57	54.00
c = 6	FE	44754.77	50832.16	3896.01	2181.38	40.39	35.41	45.80
	CB	49421.64	57066.62	6734.86	910.12	50.56	35.20	52.20
c = 6.5	FE	49405.70	55879.90	4156.19	2318.01	42.93	30.70	48.80
	CB	52922.34	61383.76	7489.24	972.18	54.01	35.65	57.20
c = 7	FE	48595.06	55259.37	4279.24	2385.07	44.17	34.81	47.60
	CB	51183.03	60079.34	8025.81	870.50	48.36	33.17	51.00

C

Branch-and-Price and Benders decomposition framework

A natural extension of column generation is Branch-and-Price, wherein the generation of new assortments is embedded within the branch-and-bound process to ensure integrality. At each node of the branch tree, one solves a restricted master problem (RMP) over a subset of columns. Afterward, a pricing subproblem identifies promising columns with positive reduced cost, which are then introduced into the RMP. When rider decisions must also be considered, a Benders decomposition can be added to separate rider from assortment selection. Algorithm 5 sketches a possible algorithmic framework:

Algorithm 5 Branch-and-Price with Benders Decomposition

Input: Initial restricted master problem (RMP) with a limited column set, initial Benders cuts, convergence tolerance
Output: Global optimal solution with assortments and rider decisions
Initialize: Include an initial set of columns (e.g., NBR assortments) in the RMP, and add initial Benders cuts approximating rider costs from NBR assortments
while not converged ***and*** branch-and-bound nodes remain **do**
    Solve the **RMP Relaxation** (LP) without integrality on $z_{S_q^d}$
    Extract **dual information** for the Pricing Problem
    Solve Pricing Problem for each (q, d):
        Identify columns with positive reduced cost
        Add any such columns to the RMP and re-solve
    if no new columns *with positive reduced cost exist* **then**
        Branch-and-Price on fractional $z_{S_q^d}$:
            Resolve RMP at each branch node with column generation
            Continue until all nodes are explored or pruned
    if integer-feasible solution *is found* **then**
        Solve the **Rider Subproblem** given the current solution
        if rider cost *exceeds the expected cost* **then**
            Generate a **Benders cut** from the rider subproblem
            Add the cut to the RMP and restart column generation
return *Optimal assortments and rider decisions*

In principle, this integrated method can deliver exact solutions for the unified optimization of assortments and rider dimensioning, but it is computationally expensive. Every node in the branch-and-bound tree becomes a specialized column-generation procedure, and each integer solution requires a rider subproblem solve.