

# Courier satisfaction in rapid delivery systems using dynamic operating regions

Ramon Auad<sup>\*1,2</sup>, Alan Erera<sup>†1</sup>, and Martin Savelsbergh<sup>‡1</sup>

<sup>1</sup>School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA

<sup>2</sup>Departamento de Ingeniería Industrial, Universidad Católica del Norte, Antofagasta, Chile

## Abstract

Rapid delivery systems where an order is delivered to a customer from a local distribution point within minutes or hours have experienced rapid growth recently and often rely on gig economy couriers. The prime example is a meal delivery system. During an operating day, couriers in such a system are used to deliver orders placed at different restaurants to different customer locations. Operating a rapid delivery network is challenging, primarily due to the high service expectations and the considerable uncertainty in both demand and delivery capacity. We seek to fill a gap in the literature by considering courier satisfaction in a rapid delivery system, which may improve retention/loyalty in a highly competitive environment. Under the premise that couriers prefer to operate in relatively small geographic areas to increase their efficiency, we propose the novel concept of *dynamic courier regions*: small operating regions for couriers which can be dynamically and temporarily expanded to allow delivery capacity to be shared between neighboring regions when necessary to keep customer service performance metrics high. We propose an optimization-based rolling horizon algorithm for courier management which handles both region resizing and delivery task assignment decisions. Experimental results for realistic settings demonstrate that the proposed algorithm successfully balances customer and courier satisfaction, simultaneously achieving delivery times that are comparable to those of a single operating region and courier satisfaction metrics that are comparable to those achieved by fixed, inflexible regions.

**Keywords:** Logistics; Capacity management; Last-mile delivery; Rolling horizon; Bipartite matching; Meal delivery

**Acknowledgements:** Co-author Ramon Auad thanks the National Agency for Research and Development (ANID) for supporting his research with the scholarship program Doctorado Becas Chile 2017 - 72180404.

**Declaration of interest:** None.

---

\*Corresponding author, rauad@gatech.edu

†alan.erera@isye.gatech.edu

‡martin.savelsbergh@isye.gatech.edu

# 1 Introduction

In the last decade, the food service industry has been significantly altered by new internet-based business models. (Acosta, Inc. and Technomic, Inc., 2018). By 2015, the US market for food delivery was valued at \$11 billion and predicted to grow annually at 16% until 2023 (Morgan Stanley Research, 2017), more than three times faster than on-premise sales (Everett, 2019; Steingoltz and Picciola, 2019). By 2030, consumption of delivered meals is projected to exceed that of homemade meals, with sales reaching \$365 billion worldwide (Cheng, 2018; Union Bank of Switzerland, 2018). During the COVID-19 pandemic of 2020, however, observed demand for online meal delivery spiked more rapidly than projections, with sales reaching unprecedented levels (Shead, 2020; Yeo, 2021).

Growth in online food delivery has many explanations, including diners wanting to spend less time cooking (Bren-ton et al., 2019; Steingoltz and Picciola, 2019; Union Bank of Switzerland, 2018). Demand for food delivery has led to the emergence of online restaurant aggregators: third-party providers of digital marketplaces where diners can order from many possible restaurants. When delivery is requested, aggregators typically also provide this service using a pooled set of delivery couriers.

Operating rapid delivery networks is a challenging problem in modern logistics due to dynamism and uncertainty (Auad et al., 2020; Reyes et al., 2018b; Yildiz and Savelsbergh, 2019a). On the demand side, customers dynamically place orders to restaurants in the network, with the system often experiencing abrupt variations in order placement rates throughout the day. Moreover, diners expect their food to be delivered usually within the hour while preserving its freshness. Thus, orders must be dispatched within minutes after they are ready, reducing opportunities to consolidate deliveries to multiple diners to reduce cost. Meal delivery network operators need to dynamically match delivery capacity (provided by *drivers* or *couriers*) to demand, and the usual tradeoffs apply. Lack of delivery capacity at some times leads degradation in customer service, while too much capacity at other times can be overly costly due to an excess of underutilized couriers.

Matching couriers to deliveries in rapid delivery systems can be especially challenging since couriers often participate in these networks with significant autonomy. In many systems, couriers do not need to accept all work assigned to them. It is typically difficult to issue repositioning decisions to couriers who instead move where they believe they are most likely to find profitable future delivery tasks. Furthermore, several logistics networks are competing for the same couriers and there are few barriers to keep couriers from moving between companies (or working for multiple networks simultaneously). Thus, maintaining high levels of courier satisfaction can be just as important as customer satisfaction for these networks.

This paper considers operational decision-making for rapid delivery networks like those for prepared meals and explicitly introduces approaches to improve courier satisfaction. Customer-centric performance metrics and courier-centric performance metrics will be considered simultaneously. Couriers are primarily motivated by the income and profit they can earn within a delivery network (Asdecker and Zirkelbach, 2020). They also prefer to work in a relatively compact geographic region, likely close to their residence in order to minimize time and cost of commuting. Working in a compact area also provides familiarity advantages including spending shorter times finding parking, restaurants, and delivery locations, and recurrently interacting with the same restaurant workers, leading to overall more efficient service times and a subsequent increase in the number of completed deliveries (Zhong et al., 2007).

To simultaneously achieve both high customer service and high courier satisfaction, we propose a structure for delivery networks in larger geographic areas where the service region is partitioned into multiple smaller *courier regions* designed to reduce or eliminate the likelihood of couriers wandering over the full extent of the service region. In the context of meal delivery, these courier regions can be created by partitioning the set of restaurants and then allocating (hiring) couriers to serve orders from restaurants in a single courier region. Since such a strategy does eliminate some of the risk-pooling uncertainty management benefits of operating a larger pool of couriers over the full service region, we further propose that these courier regions be *dynamic*, with boundaries that can be temporarily altered on-demand during the operating day. In our proposed approach for meal delivery, a region is dynamically expanded by adding a number of restaurants from a neighboring region, allowing orders from those origins to be fulfilled by couriers from either the newly-expanded region or from the original region. Limiting the dynamic region changes can hopefully keep most of the courier benefits of small regions while improving courier utilization and customer service performance metrics.

Decisions related to dynamic region reshaping and the assignment of orders to couriers are proposed to be made by a rolling horizon algorithm. The algorithm utilizes ideas from bipartite matching to both construct order-courier assignment recommendations over time while also simultaneously changing some of the restaurant-region assignments (and thus the courier region boundaries).

The main contributions of this research can be summarized as follows:

- This paper is the first to develop rapid delivery system operational models that explicitly consider courier satisfaction alongside customer satisfaction metrics and studies the tradeoffs between these two types of metrics.
- We propose a dynamic courier region operating strategy for rapid delivery systems, which seeks to capture most of the courier benefits of a static regional partitioning strategy by introducing partial flexibility to balance system objectives when facing uncertainty in order demand.
- We develop a scalable two-phase matching-based rolling horizon algorithm that optimizes key operational decisions for rapid delivery system and successfully integrates dynamic region redefinition and order-courier assignments.
- We demonstrate via an empirical study using real-world data that courier satisfaction metrics can be increased without deteriorating key order service quality metrics for a fixed courier fleet size.

The remainder of the paper is organized as follows. Section 2 provides a brief survey of the related literature. Section 3 defines the studied problem, the mathematical notation, the assumptions made and the key performance metrics to measure the quality of solutions. Subsequently, Section 4 presents the rolling-horizon matching-based algorithm to solve the problem, while Section 5 provides empirical evidence of the effectiveness of the solution approach. Lastly, Section 6 summarizes the work done and suggests directions for future research.

## 2 Literature review

Management of rapid delivery systems requires solving decision problems such as those modeled by dynamic vehicle routing problems; a comprehensive survey of the extensive literature of this class of decision models and

optimization problems can be found in [Pillac et al. \(2013\)](#) and [Psaraftis et al. \(2016\)](#). Furthermore, assignment of couriers to demands in rapid delivery can be classified in the dynamic pickup and delivery problem sub-class (dPDP), which has gained considerable research attention in over the years mostly thanks to the emergence of same-day delivery services, e-commerce, and ride-hailing applications ([Agatz et al., 2012](#); [Archetti and Bertazzi, 2021](#); [Berbeglia et al., 2010](#); [Savelsbergh and Van Woensel, 2016](#)).

Like meal delivery, many dPDPs share multiple key characteristics that complicate their solving process, such as limited knowledge of future events (*e.g.*, request placements) and tight service constraints (*i.e.*, urgency of decisions). However, researchers and practitioners have found that myopic matching-based rolling horizon algorithms can produce high-quality solutions in many application contexts and for very large-scale problem instances ([Arslan et al., 2019](#); [Dayarian and Savelsbergh, 2020](#); [Guo et al., 2018](#); [Reyes et al., 2018b](#); [Wang et al., 2018](#)). This appears to be true especially when the opportunities for consolidating multiple orders into single trips are limited and the urgency to dispatch trips and deploy capacity outweighs the benefits of reserving capacity for future uncertain requests.

Motivated by these technological developments of applications of dPDP, many authors have studied this class of problems both from a theoretical standpoint ([Archetti et al., 2015](#); [Auad et al., 2020, 2022](#); [Reyes et al., 2018a](#); [Ulmer et al., 2019](#); [Yildiz and Savelsbergh, 2019b](#)) and in more realistic settings ([Dayarian and Savelsbergh, 2020](#); [Klapp et al., 2018](#); [Reyes et al., 2018b](#); [Ulmer et al., 2021](#)). In the particular case of meal delivery applications, seminal work include the one by [Reyes et al. \(2018b\)](#), which formally introduces the meal delivery routing problem (MDRP), capturing the most crucial aspects of dPDP observed in the meal delivery context. To solve the dynamic version of the problem, the authors propose a framework based on a rolling horizon algorithm that repeatedly solves a matching to make dispatching decisions. The work provides an exhaustive computational study that analyzes the impact of a wide range of operational factors, including the complexity of the dispatching technology, the bundling level, the assignments commitment policy, and specific algorithmic features. In a parallel research effort, [Yildiz and Savelsbergh \(2019a\)](#) attempt to solve the static version of the MDRP to determine the value of having complete information about order arrivals. Due to the difficulty of solving large-scale integer programming models for these problems, they develop a solution framework that simultaneously generates model rows and columns. By solving a subset of instances from [Reyes et al. \(2018b\)](#), the combined results of both groups of authors determine that in general, a myopic rolling horizon approach is capable of finding near-optimal solutions in terms of service quality.

In related work, [Liu \(2019\)](#) study the operations of a meal delivery system where couriers are replaced by drones, considering many context-specific constraints that consider features such as order type, drone carrying capacity, and drone battery capacity. The problem is solved using traditional mixed-integer programming techniques embedded in a rolling horizon algorithm. In addition to considering the typical features defining the MDRP, our work in this paper additionally considers defining subregions of the service area to which delivery supply resources (couriers) are allocated in advance, and the management of these courier regions over time. A poor implementation of such an approach could have adverse effects on system performance metrics, since the risk pooling benefits of flexible supply resources may disappear leading to high expected costs ([Huang et al., 2019](#)) A proper region definition and allocation of resources, on the other hand, allows to better balance workload among delivery suppliers [Carlsson and Devulapalli \(2012\)](#). Managing courier regions over time with dynamic adjustments, as we propose, is in itself a complex stochastic resource allocation problem. Although many dynamic adjustment schemes are possible, the research in this paper considers a relatively simple scheme in which couriers are allocated *a priori* to regions, but

these base courier regions may be expanded temporarily during the operating day to include additional demand (in meal delivery, originating from restaurants). Demand in expanded regions can be served by couriers either from the original base region or from the (now overlapping) expanded region. Given this structure, determining how and when to expand regions or revert them to their base configuration is another stochastic optimization problem.

The dynamic courier region expansion scheme proposed is an approach to introduce *partial flexibility* to the dynamic delivery resource allocation problem. The concept of partial flexibility has been widely explored in manufacturing. [Jordan and Graves \(1995\)](#) introduce the concept of “*chaining*” in the context of flexible manufacturing and empirically show, through a simulation analysis, that slight improvements in process flexibility (*i.e.*, establishing a *long chain* between resources and jobs) are sufficient to obtain performance comparable to the ideal fully flexible process. In a subsequent paper, [Simchi-Levi and Wei \(2012\)](#) characterize the performance of long chains by first proving the supermodularity of the marginal benefits of additional flexible arcs until the long chain is formed, and then using this property to show that the long chain maximizes the system performance among all the possible 2-flexible designs.

Compared to the field of manufacturing, partial flexibility has received little attention in logistics and rapid delivery systems ([Ledvina et al., 2020](#); [Lyu et al., 2019](#)). The work by [Ledvina et al. \(2020\)](#) studies flexibility in the context of vehicle routing with stochastic demand. Although they address a different problem than ours, their solution also seeks to balance the system performance with employee (couriers) familiarity with their operated routes, by combining fixed and re-optimized delivery plans. They add partial flexibility to the system in the form of *a priori* overlapping routes (*i.e.*, routes that share some of the visited customers), and demonstrate that limited route overlapping may attain an overall performance comparable to the most flexible (and more complicated) case of fully reoptimized routes.

### 3 Problem definition

This section presents a model for optimizing rapid pickup-and-delivery operations where orders become known only shortly before they are ready for dispatch and need to be delivered to customers quickly by a target time; meal delivery is the canonical application example. The model captures the most essential aspects of real-world rapid delivery processes, namely, (i) dynamic order arrivals: *orders* are continuously placed to the system, with no advance information and ready times only minutes after placement times; (ii) multiple pickup locations: many origin locations (*restaurants*) exist, and each order must be delivered from a specific origin; and (iii) dynamic delivery capacity: to complete the delivery of orders, the system employs *couriers* and each is available during some scheduled period of time during the operating day known as their *block*. Furthermore, the model assumes that couriers are operating in the service area, and that they may be restricted to pick up orders from pickup locations (restaurants) only within specific regions of the service area at certain times.

Suppose that each courier is assigned *a priori* to a subregion within the service area, which we refer to as their *base courier region*. The courier begins their block in their base region and would prefer to operate within it for most of their block. Base courier regions are defined by partitioning the pickup locations, assigning each to a single region in such a way to form contiguous and compact geographical regions. A courier operating in region  $i$  can only serve demand originating in region  $i$ . While in principle, this definition is most appropriate for systems where delivery locations are relatively close to pickup locations (like meal delivery), it is possible to generalize these ideas.

Such a system configuration prevents couriers from roaming too far away from their base courier region and also incentivizes them to return to that region when idle to maximize the likelihood of matching with a future order; these features can both improve courier satisfaction but also balance resource availability geographically over time. However, it is also possible that service quality may degrade when couriers are partitioned into regions due to a loss of flexibility in order-courier assignments. Therefore, we also allow the decision maker to dynamically and temporarily expand courier regions when doing so will better balance supply and demand. So-called *expanded courier regions* always include the base region pickup locations and then some additional pickup locations in neighboring base regions. Using an expanded region provides the flexibility to share courier resources, and sharing may improve customer service metrics (*i.e.*, reducing delivery times) at the cost of degrading some courier satisfaction metrics (by increasing travel outside of the base region).

In this study, we model dynamic region reshaping using a discrete approach where each base region can be expanded into one or more neighboring regions by adding pre-defined sets of restaurants in the boundary area and may be contracted by removing these expansions. Thus, at various decision epochs, the decision maker needs to choose which regions to operate in their base configurations and which to operate in various expanded configurations.

### 3.1 Notation and main assumptions

Formally, let  $P$  be the set of pickup locations comprising a service area, with each  $p \in P$  having an associated two-dimensional location  $\ell_p$ . Let  $O$  be a set of orders, each order  $o \in O$  associated with a pickup location  $p_o$ , an announcement time (*i.e.*, placement time)  $a_o$ , a ready time  $r_o$ , a two-dimensional delivery location  $\ell_o$ , and due time  $d_o$ . Delivery tasks are completed by a set of couriers  $C$ , where each courier  $c \in C$  is characterized by a start location  $\ell_c$ , a block start time  $s_c$ , and a block finish time  $f_c$ . We assume that all courier information is known in advance, but that order information is known only at the time of announcement. Thus, information about the existence of order  $o \in O$  becomes available only at time  $a_o$ , along with all of the detailed information about the order, including its ready time  $r_o$ .

Suppose that the system operating day is defined by the time horizon  $\mathbf{T} \doteq [0, T]$ ,  $T > 0$ . Given  $t \in \mathbf{T}$ , let  $O(t) \subseteq O$  be the set of active orders at  $t$ , *i.e.*, placed orders that have not been delivered by time  $t$ , and let  $U(t) \subseteq O(t)$  be the set of active orders not assigned to any courier by time  $t$ . Let  $C(t) \subseteq C$  be the set of active couriers at time  $t$ , *i.e.*, couriers  $c$  such that  $t \in [s_c, f_c]$ .

The set of base courier regions is denoted by  $R \doteq \{r_1, \dots, r_m\}$ ,  $m \geq 1$ . Each base region  $r \in R$  is defined by a fixed set of pickup locations  $P_r \subseteq P$ ; the sets  $\{P_r\}_{r \in R}$  form a partition of  $P$ . Furthermore, suppose that expansion sets  $P_{r,r'}$  are also defined in advance; an expansion set  $P_{r,r'}$  contains pickup locations  $p \in P_{r'}$  that are to be added to base region  $r$  in case that region is expanded in the direction of  $r'$ . We describe a specific approach for defining  $P_{r,r'}$  in Section 4.1, but any reasonable approach for defining locations in  $r'$  that are near to  $r$  could be used. At a given time, a region may be altered via expansion or contraction actions, changing the set of pickup locations enclosed within its boundaries. Hence, we denote the set of pickup locations covered by region  $r \in R$  at time  $t \in \mathbf{T}$  as  $P(r, t)$ , satisfying  $P_r \subseteq P(r, t)$ ,  $\forall r \in R, \forall t \in \mathbf{T}$ . If  $P(r, t) \cap P_{r'} \neq \emptyset$  for  $r' \neq r$ , then it is said that  $r$  is *supporting*  $r'$  at time  $t \in \mathbf{T}$ , or equivalently, that  $(r, r')$  form a *support pair*. While a courier region is supporting another, couriers initially assigned to operate in the former become eligible to deliver orders from a subset of the latter's pickup locations. We will sometimes refer to the area of a courier region  $r$  at time  $t$ , which will be defined

as the area of the convex hull of the pickup locations  $P(r, t)$ .

Courier regions are used to control the operations of couriers. To do this, we assign to each courier  $c \in C$  a base region  $b_c \in R$  at their start time  $s_c$ . Furthermore, for  $\theta \geq 0$ , we define the *terminal period* of courier  $c$  as the last  $\theta$ -duration portion of their block  $(f_c - \theta, f_c] \subseteq [s_c, f_c]$ , and the prior portion of their block  $[s_c, f_c - \theta] \subseteq [s_c, f_c]$  as their *regular period*. Then, in our setting, courier  $c$  is eligible to pick up orders from the (possibly expanded) set  $P(b_c, t)$  during their regular period  $[s_c, f_c - \theta]$  and is eligible to pickup orders from the set  $P_{b_c}$  during their terminal period  $(f_c - \theta, f_c]$ . If at time  $t^{\text{assignment}}$  a courier  $c$  is considered for assignment to an order  $o$ , a pickup time  $t^{\text{pickup}}$  is determined which is the later of the courier arrival time to the pickup location and the order ready time  $r_o$ . If  $t^{\text{pickup}}$  falls within the regular period of  $c$ , an assignment of  $c$  to  $o$  is allowed if the pickup location  $p_o$  is in the pickup location set  $P(b_c, t^{\text{assignment}})$ . If  $t^{\text{pickup}}$  is to occur after the regular period but before the finish time  $f_c$ , an assignment is allowed only if  $p_o \in P_{b_c}$ , the base region for the courier. Note that there are no restrictions on the drop-off time and the drop-off location; the drop-off time may occur after  $f_c$ , the end of the block of courier  $c$ .

Travel times between locations are encoded in a matrix  $\tau$ . In addition, a courier that arrives at a pickup location to pick up an order incurs a total service time of  $2s^{\text{pickup}}$  ( $s^{\text{pickup}}$  minutes to walk from their vehicle to the pickup location, and  $s^{\text{pickup}}$  additional minutes to walk back). Likewise, the courier incurs a total service time of  $2s^{\text{drop-off}}$  at a drop-off location when delivering an order ( $s^{\text{drop-off}}$  minutes to walk from the vehicle to the delivery location, and another  $s^{\text{drop-off}}$  minutes to walk back to their vehicle). We assume that these times are deterministic and invariant over the operating horizon  $T$ .

To illustrate the pickup and drop-off process, consider a courier  $c \in C$  idle at some location  $\ell$  that at time  $t \in T$  is assigned to deliver order  $o \in O$ . The *pickup* of  $o$  by  $c$  occurs as soon as (i)  $o$  is released at the pickup location; and (ii)  $c$  arrives at the pickup location  $p_o$  and picks up the order, *i.e.*, exactly at time  $t^{\text{pickup}} = t + \max\{r_o, \tau_{\ell, p_o} + s^{\text{pickup}}\}$ . Once  $o$  is picked up,  $c$  departs from the pickup location  $p_o$  at time  $t^{\text{departure}} = t^{\text{pickup}} + s^{\text{pickup}}$  to deliver  $o$  at drop-off location  $\ell_o$ . The *drop-off* of order  $o$  occurs at time  $t^{\text{drop-off}} = t^{\text{departure}} + \tau_{p_o, \ell_o} + s^{\text{drop-off}}$ , and  $c$  subsequently departs at time  $t^{\text{drop-off}} + s^{\text{drop-off}}$  to continue their operation. After completing the delivery of an order, if courier  $c$  has already been assigned to a new order, then they immediately depart toward the pickup location for the new order. Otherwise, courier  $c$  repositions by traveling to the pickup location that is the closest (among locations in  $P(b_c, t)$  if the drop-off of  $o$  occurred during the regular period of  $c$ ; and among  $P_{b_c}$  if the dropoff took place during their terminal period) to the location of their last delivery  $\ell_o$ . Nonetheless, at any time during the repositioning, courier  $c$  may have their course of action modified if assigned to serve a new order.

For simplicity, suppose that couriers are compensated using a base pay per time (and thus a fixed compensation given the block duration). We further assume that couriers accept all assigned orders. Other compensation schemes are possible and could be modeled, for example those where couriers are paid per order delivered and possibly with extra compensation for orders delivered further from their origin pickup locations.

In this paper, we further restrict the problem to one where each courier can deliver only one order at a time from a pickup location. In some scenarios, there may exist operational benefits from consolidating multiple orders into a single delivery route (also known as *order bundling*); clearly, couriers may be able to complete more deliveries per time with bundling, but individual orders may be delivered with larger delays from their placement and ready times. This restriction is made primarily to simplify the analysis; the methodology that we propose for this problem can accommodate order bundling readily if necessary.

### 3.2 Performance metrics

We measure the satisfaction of couriers using the following metrics:

- **First-to-last location travel time (FtL)**: travel time between a courier’s start location and their end location.
- **First-to-furthest location travel time (FtMax)**: direct travel time between a courier’s start location and the furthest location contained in their complete route.
- **Base region order percentage ( $\mu_{base}$ )**: fraction of orders assigned to a courier during a shift with a pickup location that is in the courier’s base region.

Couriers may feel more satisfied if (i) they are allocated to a courier region that is close to their starting location, and (ii) they are asked to leave their region infrequently and do not venture far. We decide to capture these ideas using metrics that measure how many orders couriers pick up outside their base region, and how far couriers venture from their start locations. These metrics allow comparison to configurations with varying numbers of courier regions of different sizes.

Improving courier satisfaction, however, may lead to sacrifices in customer service quality since restricting the service area of couriers also reduces the number of feasible order-courier assignments. Hence, we also consider standard customer satisfaction metrics in the existing rapid delivery literature, such as:

- **Percentage of orders delivered**; an order is considered lost (not delivered) if it cannot be delivered at or before the end of the operating horizon,  $T$ .
- **Click-to-door time (CtD)**: difference between the delivery time of an order and its placement time.
- **Ready-to-door time (RtD)**: difference between the delivery time of an order and its ready time.
- **Ready-to-pickup time (RtP)**: difference between the pickup time of an order and its ready time.

When dynamically reshaping regions, a primary goal is to decrease the workload in courier regions that are overloaded with orders. To measure the workload of region  $r$  at time  $t$ , we use **order-per-courier ratio (OPC)**, denoted  $OPC(r, t)$ , which captures the number of active orders (in some time period) and the number of available couriers (in some time period). Using OPC to measure workload has two primary advantages: it can be computed easily, and it can be interpreted easily. Expanding a courier region  $r$ , i.e.,  $r$  supporting some other region  $r'$ , implies that couriers assigned to  $r$  become eligible to serve orders from some pickup locations in  $r'$  in addition to pickup locations from  $r$ . This implies that more orders are to be distributed among couriers from  $r$ , hence expanding  $r$  increases  $OPC(r, t)$ . The benefit of such an action lies in having more resources to serve orders from  $r'$ , which decreases  $OPC(r', t)$ . If at some point after the expansion of  $r$ , the performance of  $r'$  improves enough so that  $r'$  no longer needs support from  $r$ , then it is beneficial to contract  $r$  to prevent its couriers from unnecessarily roaming outside their region of choice. The computational study in this paper will demonstrate that the specific OPC metric that we compute successfully captures the load placed on the available delivery resources in a region and thus guides appropriate load-balancing through the region re-shaping mechanism.

## 4 A two-stage matching-based rolling horizon algorithm

In this section, we propose an algorithm for assigning couriers to delivery tasks and re-shaping courier regions dynamically over time. Other research has found that given the highly dynamic nature of many rapid delivery order processes (such as those for meal delivery) and the urgency of deliveries, operations may not be substantially improved by approaches that attempt to use predictive information about future order arrivals when making delivery task assignment decisions (Reyes et al., 2018b). Thus, we propose a rolling horizon algorithm that relies only on known information for delivery task assignments. We furthermore extend the framework of the approach to decide on courier region reshaping over time. At each decision epoch, a set of proposed order-to-courier assignment decisions are determined, but the decisions to be immediately implemented are determined using a so-called *commitment rule*; orders and couriers matched in uncommitted assignments are simply unmatched and reconsidered in the pool by the algorithm again after the horizon rolled for the next decision epoch. Like order-to-courier assignment, we also use ideas from bipartite matching to decide how to shape the courier regions at each decision epoch.

More specifically, the algorithm solves a set of matching optimization problems every  $f$  minutes. At decision epoch  $t$ :

- A boundary redefinition step first uses a bipartite matching optimization model to decide whether to expand some regions in  $R$  in order to support other regions with greater workload, maximizing the OPC reduction of supported regions. This step then uses a second bipartite matching to select a subset of active supporting pairs to terminate, maximizing the area reduction of the regions involved. This step is described in detail in Section 4.1.
- A delivery task assignment step solves a third bipartite matching which assigns orders in set  $U(t)$  to couriers in set  $C(t)$ , determining assignment recommendations that minimize freshness loss for orders. A commitment rule determines which of these recommendations are to be executed. This step is described in Section 4.2.

One of the main advantages of using matching models within the algorithm is that finding an optimal weight matching on problems of realistic scale for this application requires little computational effort. This is important in the highly dynamic environments faced by rapid delivery systems, where only a few minutes (at most) may be available to make decisions.

### 4.1 The courier region reshaping step

In this section, we describe the expansion and contraction decision process for courier regions. The action of expanding a region is performed to provide support to another region, while the action of contracting a region is taken to terminate unneeded region-to-region supporting pairs. Thus we model the decision of one region providing support to another one as a dichotomy of whether the region to provide the support should start covering a fixed subset of nearby pickup locations contained in the region to receive the support, in order to redistribute the excess of workload of the latter. Under this consideration, we model the expansion and contraction operations as a maximum weight bipartite matching between sets of regions. This simple model is able to simultaneously consider all the possible expansion (contraction) decisions and to select the corresponding actions that maximize the OPC

(area) reduction; for this purpose, the weights of matching models provide enough modeling power to capture these criteria. Additionally, the use of matching models presents a practical advantage: modeling an expansion or contraction plan as a region matching limits the number of modifications of each region to no more than one at a time, thus preventing sudden changes in the overall geographical configuration at each iteration of the rolling horizon algorithm.

For the region reshaping step, we consider the following mathematical notation. Let  $\varepsilon \geq 0$  be a travel time threshold, and let  $\gamma_r$  be the geographical center of base region  $r \in R$ , defined as  $\gamma_r \doteq \frac{1}{|P_r|} \sum_{p \in P_r} \ell_p$ . For  $r, r' \in R$ , the set of pickup locations in base region  $r'$  that are covered by  $r$  if  $r$  starts supporting  $r'$  corresponds to  $P_{r,r'} \doteq \{p \in P_{r'} : \tau_{\gamma_r, \ell_p} \leq \varepsilon\}$ . Intuitively,  $P_{r,r'}$  contains all the pickup locations of region  $r'$  that are at an acceptable travel time from region  $r$ .

#### 4.1.1 OPC computation

For a courier region  $r \in R$  and time  $t \in T$ , let  $O(r, t)$  be the set of active orders at time  $t$  at a pickup location in  $P(r, t)$ ;  $U(r, t) \subseteq O(r, t)$  be the set of orders that are still unassigned by time  $t$ ; and  $C(r, t)$  the set of active couriers at time  $t$  allocated to region  $r$ . Algorithm 1 shows the routine that computes the OPC of a region. For region  $r$  and time  $t$ , the routine first counts the number of active couriers assigned to courier region  $r$ , increasing the count by one for each courier in its regular period, and by a potentially smaller value in its terminal period, as the courier may only be eligible to serve a fraction of the orders in  $r$ , *i.e.*, only the orders in  $P_r$  (lines 3 - 7). In the case the count is zero, then  $OPC(r, t)$  is immediately assigned a large value, represented by  $\infty$  (lines 8 and 9). Otherwise, the algorithm proceeds to count the number of active orders in  $r$ , *i.e.*, the orders that are or may be assigned to a courier assigned to  $r$ : for each unassigned order  $o \in U(r, t)$ , the count is increased by a value (less than or equal to one) based on the number of regions containing pickup location  $p_o$  at time  $t$ ; for each already assigned order, the count is increased by one only if the courier assigned to it is assigned to  $r$ , as otherwise such order does not make use of the region's delivery resources (lines 10 - 14).

---

#### Algorithm 1 (OPC\_COMPUTATION)

---

**Input:** Region  $r$ , time stamp  $t$ .

**Output:** Order-per-courier ratio  $OPC(r, t)$ .

```

1:  $n_{orders} \leftarrow 0$ 
2:  $n_{couriers} \leftarrow 0$ 
3: for  $c \in C(r, t)$  do
4:   if  $t > f_c - \theta$  and  $O(r, t) \neq \emptyset$  then
5:      $n_{couriers} \leftarrow n_{couriers} + \frac{|\{o \in O(r, t) : p_o \in P_r\}|}{|O(r, t)|}$ 
6:   else
7:      $n_{couriers} \leftarrow n_{couriers} + 1$ 
8: if  $n_{couriers} = 0$  then
9:   return  $\infty$ 
10: for  $o \in O(r, t)$  do
11:   if  $o \in U(r, t)$  then
12:      $n_{orders} \leftarrow n_{orders} + \frac{1}{|\{r \in R : p_o \in P(r, t)\}|}$ 
13:   else if  $o$  is already assigned to a courier  $c \in C(r, t)$  then
14:      $n_{orders} \leftarrow n_{orders} + 1$ 
15: return  $\frac{n_{orders}}{n_{couriers}}$ 

```

---

**Region workload assessment.** To evaluate the expansion or contraction of regions, we require a classification criterion to decide whether a region is eligible to provide or receive support. Although many different rules may be defined for this purpose, we simply use a threshold on the OPC value,  $\overline{OPC}$ , to perform such region classification. Then we partition the set of (possibly already expanded) regions into the subsets  $R^+(t) \doteq \{r \in R : OPC(r, t) \leq \overline{OPC}\}$  and  $R^-(t) \doteq \{r \in R : OPC(r, t) > \overline{OPC}\}$ , which respectively define regions whose workload makes them eligible to provide support to other regions, and regions whose workload makes them eligible to receive support from other regions.

#### 4.1.2 Region expansion matching model

We model this step as a maximum weight bipartite matching problem with disjoint node sets  $R^+(t)$  and  $R^-(t)$ . Each arc represents the decision of enabling the corresponding support pair, and its weight captures the associated reduction in the OPC of the supported region. The practical motivation of this operation is to distribute the workload between regions more evenly, in particular when two or more neighboring regions have an elevated workload disparity.

Let  $t \in \mathbf{T}$  be the time at which the expansion step is performed. For each  $r \in R^+(t)$ , let  $R^-(r, t) \doteq \{r' \in R^-(t) : P_{r,r'} \neq \emptyset\}$  be the set of regions that are eligible to receive support from  $r$ . For every potential (*i.e.*, not currently active) support pair  $(r, r')$ ,  $r \in R^+(t)$ ,  $r' \in R^-(r, t)$ , let  $\Delta_{r,r'}^{OPC}$  denote the minimum of the difference  $OPC(r', t) - \overline{OPC}$  and the decrease of  $OPC(r', t)$  that would result from enabling support from  $r$  to  $r'$ . To mathematically formulate the problem, consider the following decision variables:

$$z_{r,r'} = \begin{cases} 1 & \text{if } r \text{ is selected to start supporting } r', \quad \forall r \in R^+(t), \quad \forall r' \in R^-(r, t) \\ 0 & \text{otherwise} \end{cases}$$

The expansion step solves the following optimization program:

$$\max \sum_{r \in R^+(t)} \sum_{r' \in R^-(r, t)} \Delta_{r,r'}^{OPC} z_{r,r'} \quad (1a)$$

$$\text{s.t.} \quad \sum_{r' \in R^-(r, t)} z_{r,r'} \leq 1, \quad \forall r \in R^+(t) \quad (1b)$$

$$\sum_{r: r' \in R^-(r, t)} z_{r,r'} \leq 1, \quad \forall r' \in R^-(t) \quad (1c)$$

$$z_{r,r'} \in \{0, 1\}, \quad \forall r \in R^+(t), \quad \forall r' \in R^-(r, t) \quad (1d)$$

Objective (1a) seeks to maximize the total OPC reduction of regions that start receiving support. Constraints (1b) require that each potential supporter starts providing support to at most one eligible region, and Constraint set (1c) states that each new region seeking support may start receiving assistance from no more than one new eligible supporter; for each region, we limit support changes to at most one at a time to prevent severe changes in the size of regions. For optimal solution vector  $z^*$ , any pair of regions  $(r, r')$  such that  $z_{r,r'}^* = 1$  translates into region  $r$  starting to cover some of the pickup locations from  $P_{r'}$ , *i.e.*, the pickup location set  $P(r, t)$  incorporating the set of pickup locations  $P_{r,r'}$ .

### 4.1.3 Region contraction matching model

This step also solves a maximum weight bipartite matching with disjoint node sets corresponding to region sets; however, each arc now represents an ongoing support pair, and its weight corresponds to the area reduction from severing the associated ongoing support. From a practical perspective, this procedure allows to maintain courier regions as compact as possible, by contracting regions whose expansion is no longer required to maintain the workload of some neighbors under the considered threshold, thereby preventing couriers from unnecessarily roaming outside their base courier region.

Let  $t \in \mathbf{T}$  be the time at which the contraction step is performed. For each region  $r \in R$ , let  $R^+(r, t) \subseteq R^+(t)$  be the set of regions  $r'$  receiving support from  $r$  by time  $t$ , such that if the support from  $r$  to  $r'$  is interrupted, the resulting value of  $OPC(r', t)$  would remain below the threshold  $\overline{OPC}$ . For each pair  $(r, r')$ ,  $r \in R, r' \in R^+(r, t)$  let  $\Delta_{r,r'}^{area}$  be the area reduction of region  $r$  that would result from terminating the support provided to  $r'$ . The mathematical formulation of the problem considers the following decision variables:

$$y_{r,r'} = \begin{cases} 1 & \text{if the support from } r \text{ to } r' \text{ is interrupted,} \\ 0 & \text{otherwise} \end{cases} \quad \forall r \in R, \quad \forall r' \in R^+(r, t)$$

The contraction step then solves the integer program given by

$$\max \sum_{r \in R} \sum_{r' \in R^+(r, t)} \Delta_{r,r'}^{area} y_{r,r'} \quad (2a)$$

$$\text{s.t.} \quad \sum_{r' \in R^+(r, t)} y_{r,r'} \leq 1, \quad \forall r \in R \quad (2b)$$

$$\sum_{r \in R: r' \in R^+(r, t)} y_{r,r'} \leq 1, \quad \forall r' \in R^+(t) \quad (2c)$$

$$y_{r,r'} \in \{0, 1\}, \quad \forall r \in R, \quad \forall r' \in R^+(r, t) \quad (2d)$$

Objective (2a) maximizes the total area reduction from severed supports. As in the expansion model, Constraint sets (2b) and (2c) restrict support changes in each region to at most one, to prevent drastic changes in the current region sizes. Given optimal solution vector  $y^*$ , pairs of regions  $(r, r')$  such that  $y_{r,r'}^* = 1$  translates into region  $r$  terminating the support provided to  $r'$ , *i.e.*, the set of pickup locations  $P_{r,r'}$  being subtracted from the set  $P(r, t)$ .

## 4.2 The order-courier assignment step

Given the courier region configuration resulting from the region reshaping step, this second step constructs next pickup recommendations for each courier by solving a maximum weight bipartite matching problem; the optimization problem is formulated as a bipartite graph whose node sets correspond to the set of unassigned active orders  $U(t)$  and the set of operating couriers  $C(t)$ , and where each arc represents a feasible assignment between an order and a courier. Each potential assignment has an associated weight that captures the loss of freshness of the associated order before the courier picks it up.

### 4.2.1 The matching model

Let  $t \in \mathcal{T}$  be the assignment step execution time. For each order  $o \in U(t)$ , let  $C(o, t)$  be the set of couriers that at time  $t$  are eligible to be assigned to deliver  $o$ ; and further consider the set of eligible order-courier assignments  $\mathcal{A}(t) \doteq \{(o, c) : o \in U(t), c \in C(o, t)\}$ . This step determines whether or not each unassigned order  $o \in U(t)$  should be assigned to some feasible courier  $c \in C(o, t)$ , and if so which courier it is assigned to. Thus, we make use of the following decision variables:

$$x_{o,c} = \begin{cases} 1 & \text{if order } o \text{ is assigned to courier } c \text{ for delivery, } \forall (o, c) \in \mathcal{A}(t) \\ 0 & \text{otherwise} \end{cases}$$

$$y_o = \begin{cases} 1 & \text{if order } o \text{ remains unassigned after the current execution, } \forall o \in U(t) \\ 0 & \text{otherwise} \end{cases}$$

For each feasible assignment  $(o, c) \in \mathcal{A}(t)$ , let  $t_{o,c}^{pickup}$  be the pickup time of order  $o$  if assigned to courier  $c$ . The weight associated with assignment  $(o, c)$  is then computed as  $t_{o,c}^{pickup} - r_o$ . This term corresponds to the RtP and measures the freshness loss incurred by order  $o$  while awaiting pickup if assigned to courier  $c$ , due to a potential mismatch between  $o$ 's ready time and its pickup time. Consequently, we determine the assignment recommendations by solving the following assignment problem:

$$\min \sum_{o \in U(t)} w_o y_o + \sum_{(o,c) \in \mathcal{A}(t)} (t_{o,c}^{pickup} - e_o) x_{o,c} \quad (3a)$$

$$\text{s.t. } y_o + \sum_{c \in C(o,t)} x_{o,c} = 1, \quad \forall o \in U(t) \quad (3b)$$

$$\sum_{o \in U(t): c \in C(o,t)} x_{o,c} \leq 1, \quad \forall c \in C(t) \quad (3c)$$

$$y_o, x_{o,c} \in \{0, 1\}, \quad \forall o \in U(t), \forall c \in C(o, t) \quad (3d)$$

Objective (3a) minimizes the total freshness loss of the selected assignments prior to the pickup of the associated orders; additionally, a weight  $w_o$  is associated with the decision of leaving order  $o$  unassigned, which satisfies  $w_o > \max_{c \in C(o,t)} t_{o,c}^{pickup} - r_o$  to encourage assigning as many orders as possible. Constraints (3b) require that every unassigned order is either assigned to an eligible courier or left unassigned until the next iteration of the rolling horizon algorithm. In turn, Constraint set (3c) ensures that every courier receives at most one assignment recommendation.

### 4.2.2 The recommendation confirmation criteria

To reduce the effect of uncertainty in the outcome of the assignment decisions, we employ a simple confirmation criteria that avoids executing the least urgent assignment recommendations. Let  $e_c$  be the earliest time by which courier  $c$  may start serving a new order. Then, given an optimal assignment vector  $x^*$ , for each feasible assignment  $(o, c)$  such that  $x_{o,c}^* = 1$ , the instruction of serving order  $o$  is communicated to courier  $c$  only if  $o$  becomes ready for pickup and  $c$  finishes its last scheduled assignment before the next optimization, *i.e.*, if  $\max\{r_o, e_c\} < t + f$ .

## 5 Experimental results

In this section we present the performance results obtained from applying the proposed framework to two instances from the Grubhub MDRP instance repository (<https://github.com/grubhub/mdrplib>), devised using real-world daily historic data from different metropolitan areas. In accordance to the nomenclature used by the repository, the particular instances we consider correspond to:

- $0o100t100s2p100$ : a relatively small instance with 505 orders, 116 pickup locations, and 117 couriers. We refer to this instance to as *Instance A*.
- $9o100t100s2p100$ : a relatively large instance with 1746 orders, 270 pickup locations, and 432 couriers. We refer to this instance to as *Instance B*.

Figure 1 shows the pickup locations, the order delivery locations, and start locations of couriers for each considered instance.

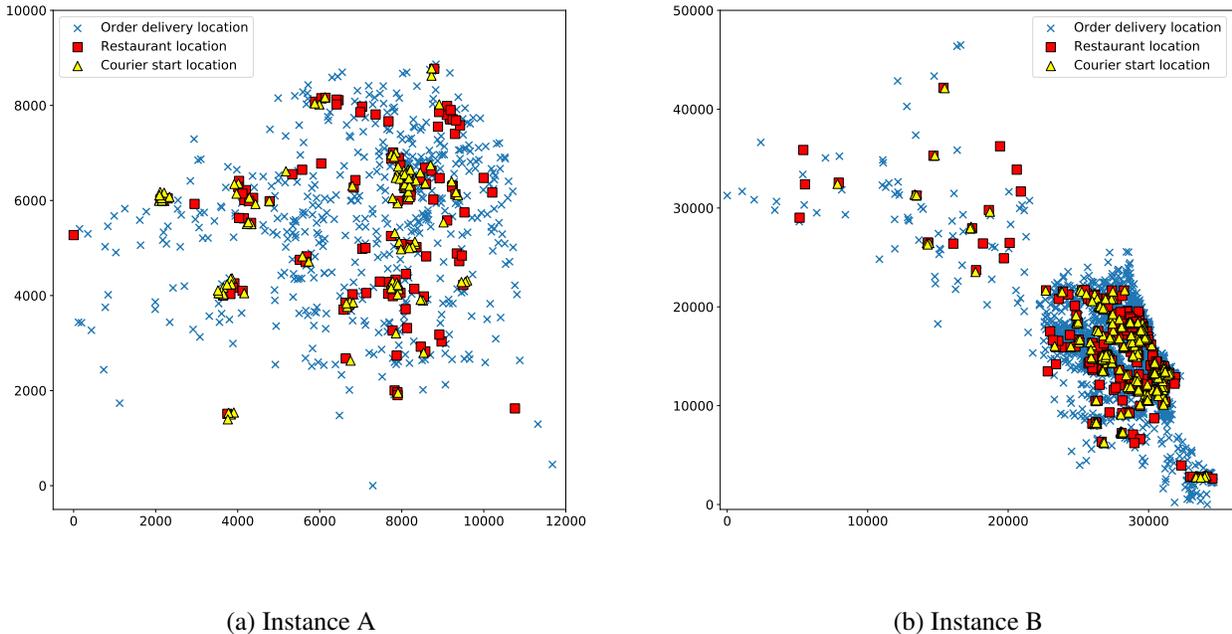


Figure 1: Visualization of instance setups

The operating horizon  $T$  represents a day-long operating period of 900 minutes. For two locations  $\ell = (x_1, y_1)$  and  $\ell' = (x_2, y_2)$ , the travel time from  $\ell$  to  $\ell'$  is computed as  $\tau_{\ell, \ell'} = \left\lceil \frac{1}{\nu} \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \right\rceil$ , where  $\nu$  is the courier's speed, set to  $\nu = 320$  meters per minute. The pickup and drop-off service times are  $s^{pickup} = s^{drop-off} = 2$  minutes, and the due time of each order  $o \in O$  is  $d_o = a_o + 40$  minutes. The rolling horizon algorithm is executed every  $f = 5$  minutes.

Each instance is run several times, each time with a different numbers of regions,  $m$ . More specifically, we run the proposed algorithms for the following configurations:

- $m = 1$ , *i.e.*, a single region;

- $m \geq 2$ ,  $\varepsilon = 0$ , *i.e.*, multiple static regions; and
- $m \geq 2$ ,  $\varepsilon > 0$ , *i.e.*, multiple dynamic regions.

The configuration with  $m = 1$  corresponds to the case where every order can be assigned to any on-duty courier, and hence its also referred to as the fully flexible case; as such, it is expected that this scenario attains the best diner satisfaction (*i.e.*, lowest CtD, RtD and RtP), although possibly having couriers roaming across the entire region, incurring an overall deterioration of the satisfaction of couriers. On the contrary, the configurations with  $m \geq 2$  and  $\varepsilon = 0$  represent the most restrictive case in terms of order-courier assignments since couriers are limited to only one of the  $m$  static base courier regions, and therefore it is expected to observe a service quality degradation. However, the fact that couriers operate in smaller localized regions should produce overall greater levels of courier satisfaction (*i.e.*, lower FtL and FtMax). The configurations with  $m \geq 2$  and  $\varepsilon > 0$  represent the hybrid approach with a set of  $m$  base courier regions to ensure courier satisfaction, but base courier regions that can temporarily be enlarged to improve system-wide delivery performance to ensure diner satisfaction. For these configurations, the parameter values  $\varepsilon$ ,  $\theta$ , and  $\overline{OPC}$  are tuned for each run.

The primary goal of the computational experiments is to demonstrate, empirically, that employing dynamic courier regions, which adds only a limited amount of flexibility to the system, suffices to achieve high satisfaction levels for both couriers and diners.

When reporting the results for a configuration in which the system is unable to deliver all the orders by the end of the operating horizon,  $T$ , we replace the CtD, RtD, and RtP values of each undelivered order by the corresponding maximum value among the orders successfully delivered in the same run, this in order to perform comparisons between configurations using the same set of orders.

The base courier region construction algorithm employed when  $m \geq 2$  can be found in Appendix A. For each instance, once the regions are constructed, each courier is allocated to the region that contains the pickup location that is closest to their start location.

## 5.1 Instance A

Figure 1a illustrates the setup of Instance A, and Figure 2 provides the resulting base courier regions for a partition with  $m = 4$  regions. The orders' pickup and delivery locations are homogeneously distributed over the considered territory. For the configurations with multiple dynamic regions, Table 1 shows the parameter values used each value of  $m$ . The values were obtained by performing a tuning process of the parameters for each value of  $m$ , using a grid

Table 1: The tuned parameter values for the different number of dynamic regions for Instance A

$m$	2	3	4	5
$\varepsilon$	25	25	25	40
$\overline{OPC}$	1.8	1.8	1.8	1.8
$\theta$	10	10	10	20

search over the ranges of values in Table 2 and selecting the combination that yields the most balanced performance in terms of CtD and FtL among all the non-dominated pairs of values. Figure 3 shows the obtained courier-related metric values for the different configurations. Naturally, increasing the number of regions,  $m$ , results in smaller base courier regions, which, in turn, results in courier routes that are more localized around their start locations.

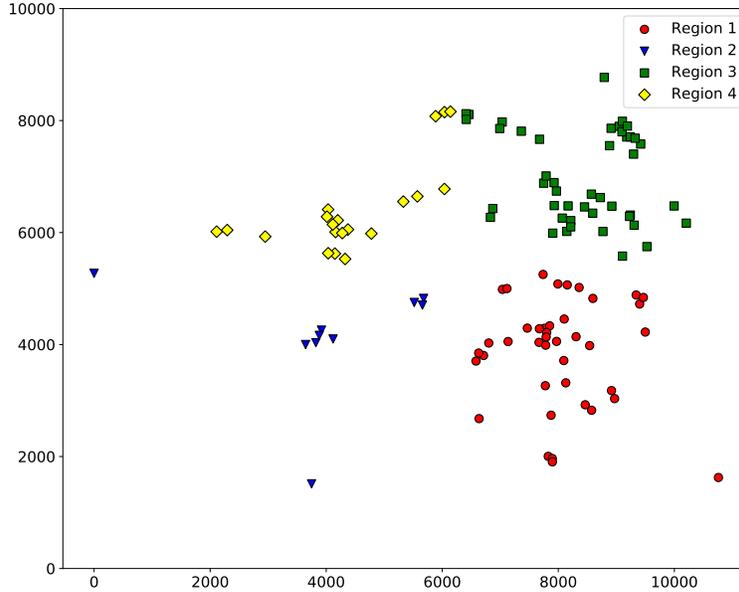


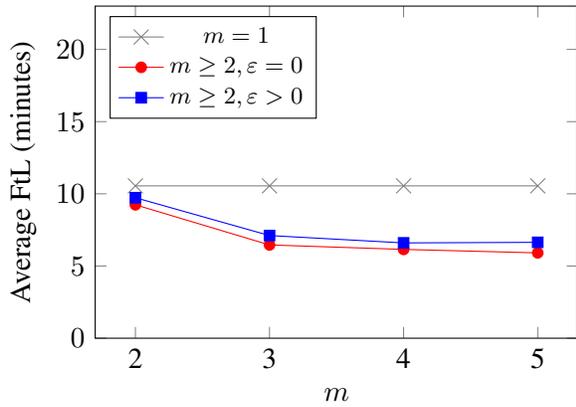
Figure 2: Partition of pickup locations from Instance A into  $m = 4$  base courier regions

Table 2: The values used for the parameter tuning for Instance A

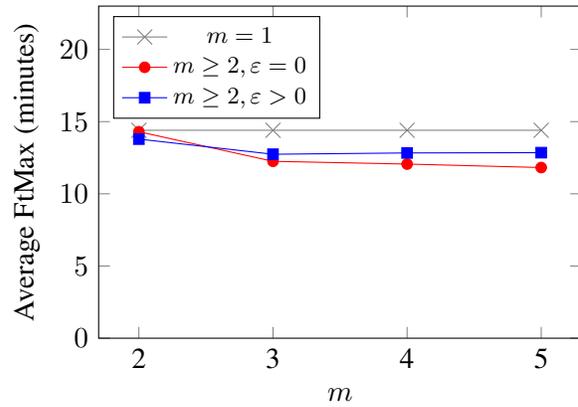
$\varepsilon$	$\overline{OPC}$	$\theta$
{5, 10, 15, 20, 25, 30, 40}	{1.0, 1.2, 1.5, 1.8, 2.0}	{0, 10, 20, 30}

This is reflected in the reductions of the average FtL and FtMax compared to the fully flexible case, up to 40% and 12%, respectively. However, smaller base courier regions also result in less flexibility when assigning orders to couriers, causing a deterioration in the system delivery performance. This is illustrated in Figure 4, which shows the average values of order-related metrics obtained for the different configurations. We see that with more than three multiple static regions the system is no longer able to serve all the orders. Even when all orders can be served, the rigidity of the setting results in a deterioration of all other delivery time metrics: across all values of  $m$ , static regions experience an average deterioration of the average CtD of 32% relative to the fully flexible setting; furthermore, for the case with  $m = 4$  static regions, the system fails to serve 1.5% of the orders and suffers an increase of the average CtD of 17.7% with respect to the case  $m = 1$ . Importantly, adding slightly more flexibility with multiple dynamic regions almost eliminates the service quality loss. Across all values of  $m$ , dynamic regions incur an average detriment of the average CtD of only 5.5% with respect to the fully flexible case; moreover, a setting with  $m = 4$  dynamic regions makes it possible to serve all orders while achieving an average CtD that is only 1.6% higher than the average CtD of the fully flexible case. The system achieves this performance by exploiting the capability of making assignments between couriers and orders outside their base region. However, we observe that for all dynamic region configurations couriers seldom are required to travel outside their base region: in particular, the fraction of orders served by a courier with a pickup in their base region,  $\mu_{base}$ , is over 80% for all the considered values of  $m$ , see Figure 3c.

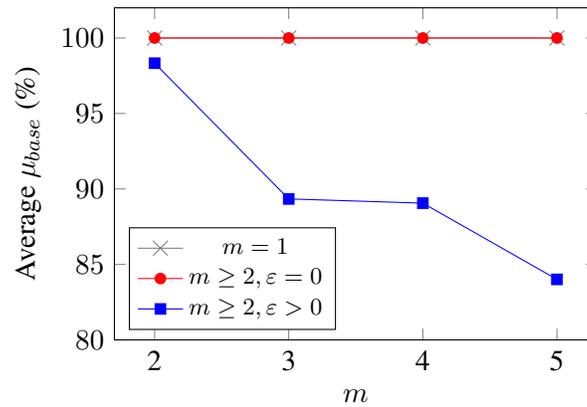
More details for the settings with  $m = 4$  regions are provided in Figure 5. We see that using dynamic courier regions achieves similar average CtD values as the fully flexible case with delivery routes that are almost as compact as those obtained when using static courier regions. The average FtL for multiple static and dynamic regions are



(a) Average FtL

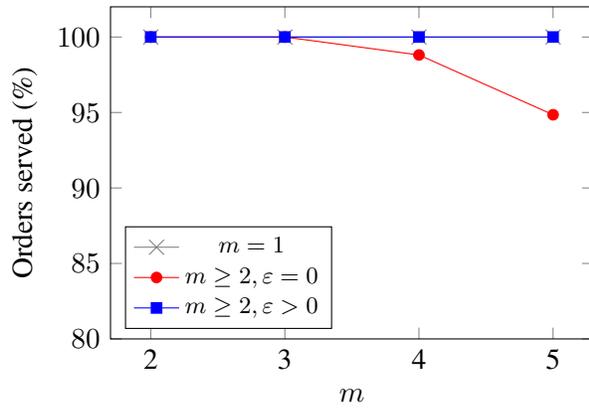


(b) Average FtMax

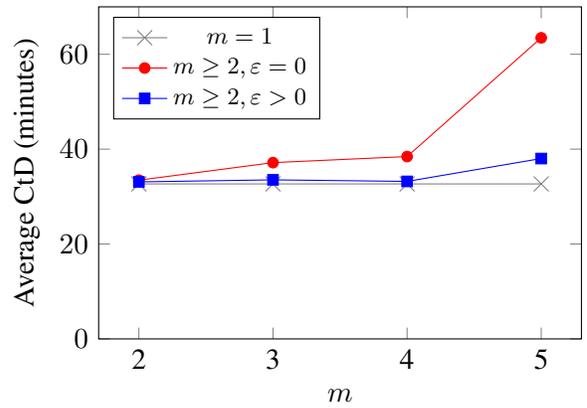


(c) Average  $\mu_{base}$

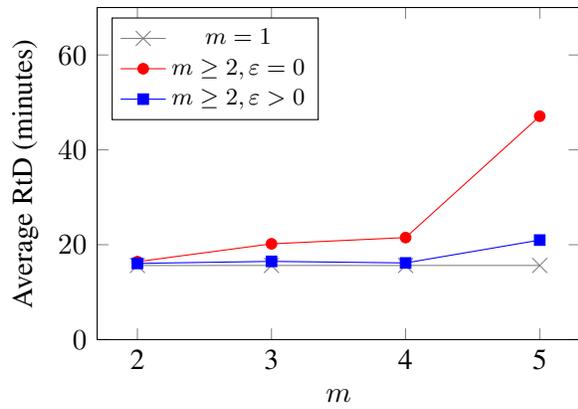
Figure 3: Comparison of courier statistics between different configurations for Instance A



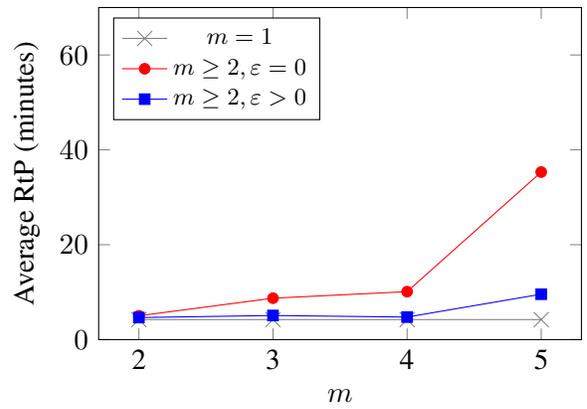
(a) Percentage of orders delivered



(b) Average CtD



(c) Average RtD



(d) Average RtP

Figure 4: Comparison of order statistics between different configurations for Instance A

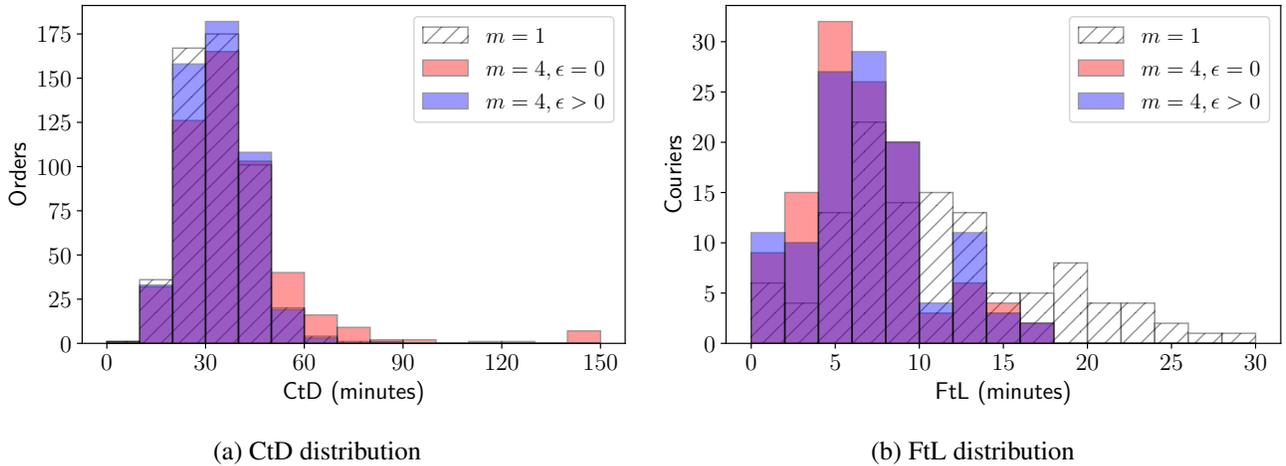


Figure 5: Distributions of order and courier satisfaction metrics for Instance A considering  $m \in \{1, 4\}$  regions

almost identical and 33% less than the average FtL of the fully flexible case. On the other hand, even though the average CtD is similar in all settings, the 90-percentile of CtD of the fully flexible case and the multiple dynamic regions setting are similar but 14.5% smaller than the multiple static regions setting.

## 5.2 Instance B

Figure 1b shows the setup of Instance B, and a 9-region partition is illustrated in Figure 6. This instance has

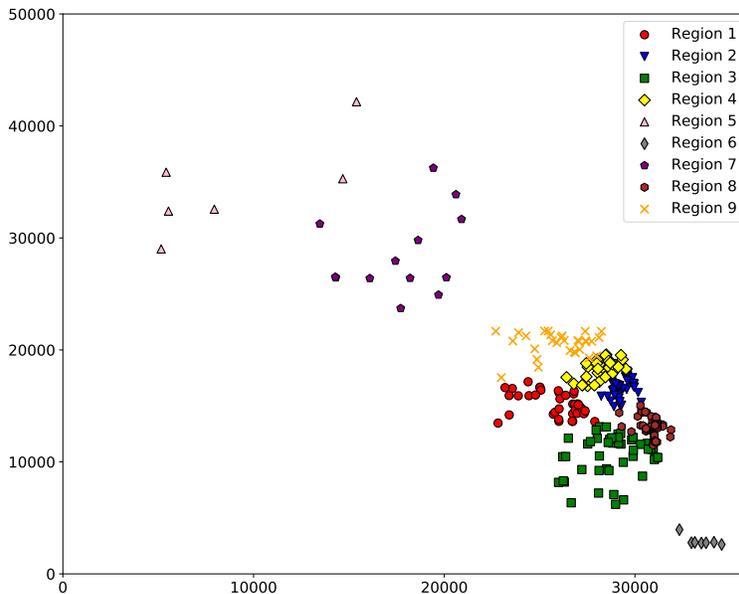


Figure 6: Partition of pickup locations from Instance B into  $m = 9$  base courier regions

a considerably larger order volume and pickup and delivery locations are no longer homogeneously distributed across the service area. The parameter values used for the configurations with multiple dynamic regions are listed in Table 3, which we obtain via parameter tuning (similarly to Instance A) using the value sets in Table 4. Despite the differences in geography and order volume compared to Instance A, the findings for Instance B are

Table 3: Parameter values used for simulating different number of dynamic regions for Instance B

$m$	2	3	4	5	6	7	8	9	10	11	12
$\varepsilon$	60	70	70	70	60	50	50	50	50	50	60
$\overline{OPC}$	1.0	1.2	1.5	1.2	1.2	1.5	1.8	1.8	1.8	1.8	1.8
$\theta$	20	0	30	0	20	10	0	10	0	0	20

Table 4: Sets of values employed in the parameter tuning for Instance B

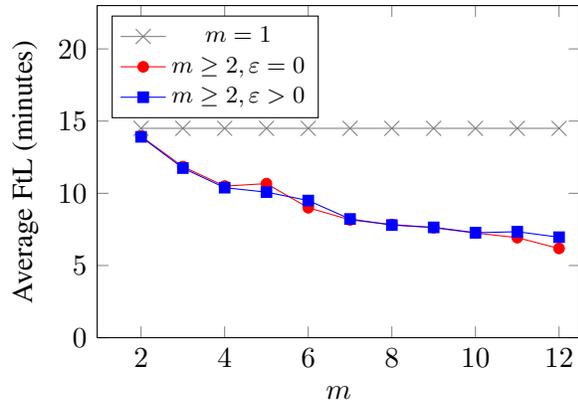
$\varepsilon$	$\overline{OPC}$	$\theta$
{10, 20, 30, 40, 50, 60, 70, 80}	{1.0, 1.2, 1.5, 1.8, 2.0}	{0, 10, 20, 30}

similar in terms of both courier satisfaction and service quality when using dynamic courier regions. The courier satisfaction metrics for the different configurations are shown in Figure 7. As before, we see that increasing the number of courier regions results in a decrease in both the average FtL and FtMax compared to the fully flexible case. Interestingly, the average FtL values when using either multiple static or multiple dynamic regions are nearly identical, but are significantly smaller than the average FtL values for the fully flexible case, especially when the number of regions gets larger (a reduction of more than 50% for some configurations).

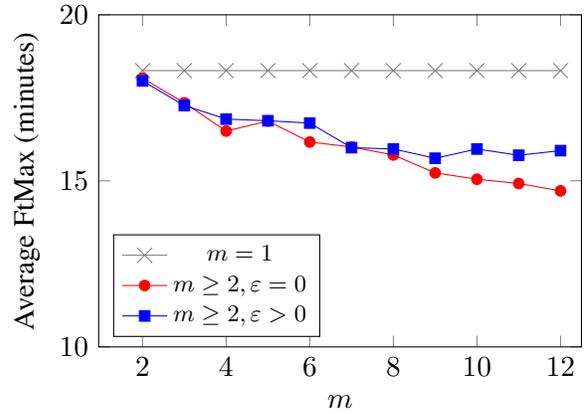
For configurations with nine or more regions, the reduction in average FtMax values compared to the fully flexible case is slightly less with dynamic regions than with static regions. However, for these configurations using dynamic regions still allows every order placed to be delivered (thanks to the option to assign couriers to orders with a pickup outside their base region, as illustrated in Figure 7c), whereas that is no longer possible using static regions (see Figure 8a). That is, the limited flexibility provided dynamic regions suffices to be able to serve every order; the price paid for this limited flexibility is small: an increase in the average CtD of less than 6% (which is acceptable considering the reductions in average FtL and FtMax). The results in terms of order-related metrics are even better when the number of regions is less than nine, as shown in Figure 8, with average CtD values increasing less than 1.5% when using dynamic regions.

More details for the settings with nine regions are provided in Figure 9. We see that the distribution of CtD values is similar in all settings, except for the fact that with static regions it is not possible to serve every order (10 out of 1746 orders are lost). However, the differences are more pronounced when focusing on the upper 1.5% of the observed CtD values: for these orders, the CtD values are 32% larger when using static region rather than allowing full flexibility, whereas the CtD values are only 6% larger when using dynamic regions rather than allowing full flexibility.

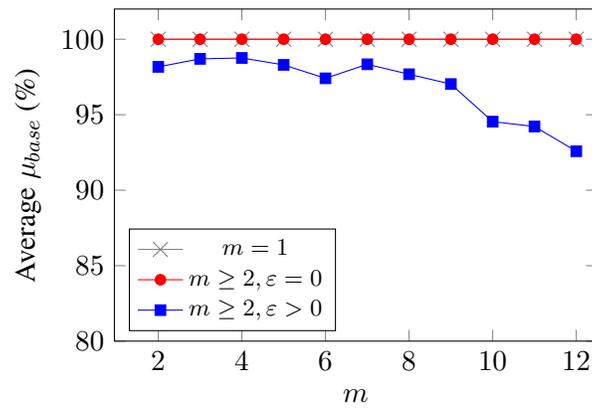
Importantly, the courier satisfaction metric, i.e., average FtL value, is much better when using dynamic regions. Using dynamic regions results in an average FtL value that is 14% lower than when using static regions and 54% lower than when allowing full flexibility. The benefit of using dynamic regions rather than static regions sometimes manifests itself because the start location of a courier of an expanded region is closer to the pickup location of an order from the region receiving support than the locations of its own couriers. In that case, using the courier from the expanded region might result in better courier satisfaction than assigning the order to a courier from the supported region. Lastly, the 95-percentiles of FtL values for the configurations with static and dynamic multiple courier regions are identical and 47% lower than the value observed with full flexibility.



(a) Average FtL

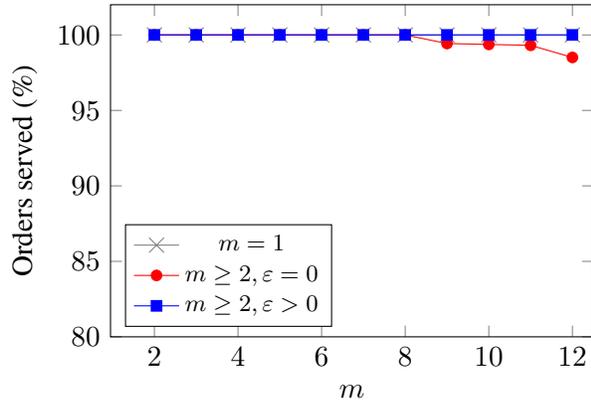


(b) Average FtMax

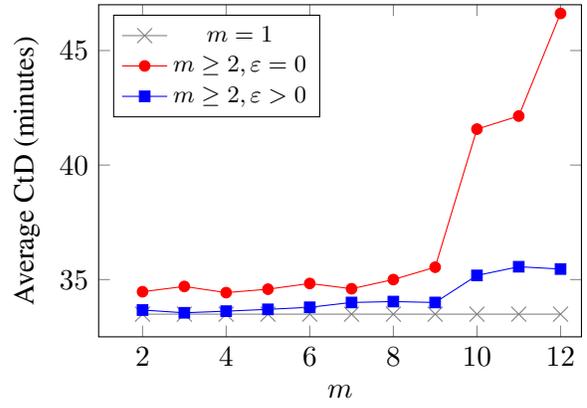


(c) Average  $\mu_{base}$

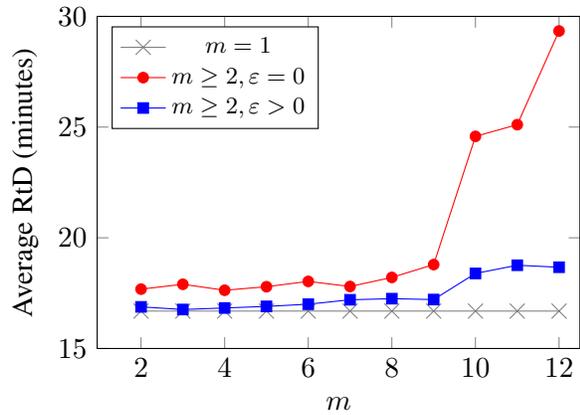
Figure 7: Courier satisfaction statistics for the different configurations for Instance B



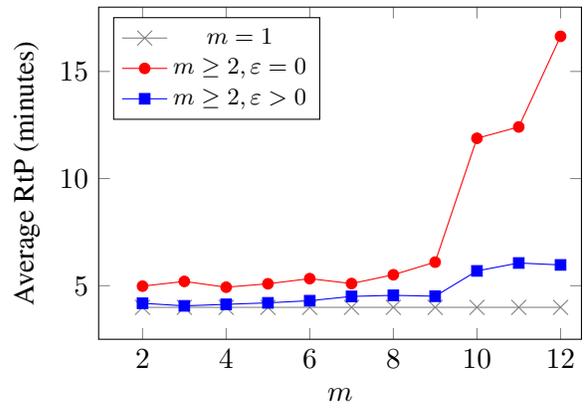
(a) Percentage of orders delivered



(b) Average CtD

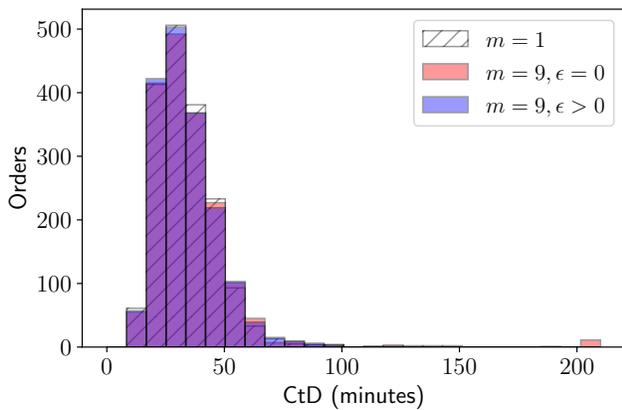


(c) Average RtD

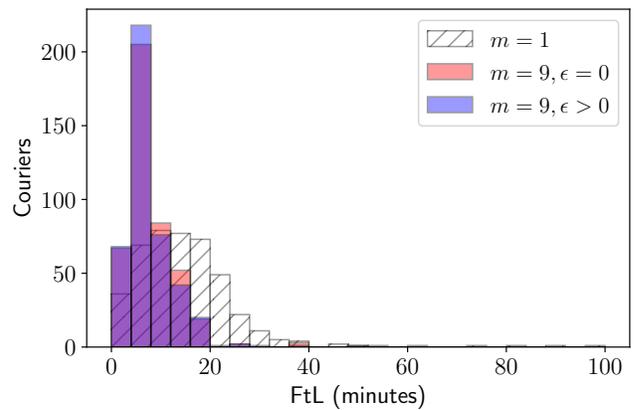


(d) Average RtP

Figure 8: Service quality statistics for the different configurations for Instance B



(a) CtD distribution



(b) FtL distribution

Figure 9: Distributions of order and courier satisfaction metrics for  $m \in \{1, 9\}$  regions

### 5.3 Discussion

The results for Instance A and Instance B have shown that the additional flexibility of dynamic courier regions makes it possible to attain diner satisfaction levels that are similar to those that are seen with full flexibility (*i.e.*, a single region), while achieving courier satisfaction levels comparable to those that are seen static courier regions. This is practically relevant as it implies that when the platform posts blocks to be booked by couriers, the platform can specify not only the block start and end time, but also the operating area – where the courier will operate for most (if not all) of the block. This additional information may be appreciated by the couriers.

Another advantage of dynamic regions over static regions is that it simplifies the planning process. With static regions, it is extremely challenging for a planner to determine an effective set of courier blocks for each of the regions. As this involves predicting order volumes and order placement patterns, the resulting (rigid) courier allocations may result in a degradation in system performance if there are unexpected changes in demand, *i.e.*, increased costs due to undelivered orders and/or lower courier utilization. This risk, however, is greatly reduced when using dynamic regions, as sharing delivery resources between regions dynamically can correct imbalances in their workloads and can mitigate the negative impact of deviations from demand predictions. This is seen (to some extent) in the experiments: dynamic courier regions allow serving of all orders when this is no longer possible with static regions (with little decrease in service quality compared to a fully flexible setting).

## 6 Conclusion

Our research represents the first attempt at incorporating courier satisfaction in the planning of meal delivery operations, under the assumption that couriers prefer to operate in a relatively small, local area. Our computational experiments show that courier satisfaction can be increased with minimal impact on diner satisfaction. We achieve this by introducing the notion of dynamic courier regions: small operating regions to which couriers are allocated and whose boundaries can be enlarged in an on-demand fashion. Small operating regions improve courier satisfaction whereas dynamically enlarging these regions mitigates (most) of the potential negative impact on service quality.

We introduce a rolling horizon approach, which repeatedly solves a sequence of bipartite matching problems to determine both the active region boundaries and the order-courier assignments, for managing dynamic courier regions. Using the approach to solve real-world instances of different sizes and geographical distributions, we have demonstrated its effectiveness: the use of multiple small regions with dynamic boundaries excels at producing well-balanced solutions, with service quality levels on-par with operating a single large region, and courier satisfaction levels akin to the use of small invariant regions.

There are several research directions that are worth further exploration. An intuitive next step is to evaluate the benefit of a small number of “unrestricted” couriers, *i.e.*, a small number of couriers that can serve orders from any restaurant; in practice, some couriers may be willing to operate in broader areas, and so it might be possible to attain even better service quality by exploiting that fact. Another natural next step is to incorporate more complex behavioral modeling of couriers. This may have important performance repercussions in the operation of meal delivery systems and to the best of our knowledge, it is a topic yet to be studied. Lastly, an alternative research direction involves the use of learning-based methods to perform dynamic reshaping of regions. An alternative

to using a rolling horizon algorithm is to train a deep neural network to construct a region resizing policy via deep reinforcement learning techniques. Then for a given state of the system, the resulting policy would dictate the modifications of the courier regions. Such an approach might be able to better capture stochastic aspects of meal delivery operations in the decision making process, possibly achieving superior levels of courier and diner satisfaction.

## References

- Acosta, Inc. and Technomic, Inc. (2018). The why? behind the dine™. Retrieved from <https://www.acosta.com/news/convenience-is-key-for-us-diners-according-to-acosta-and-technomic-s-the-> Accessed April 2022.
- Agatz, N., Erera, A., Savelsbergh, M., and Wang, X. (2012). Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2):295–303. <https://doi.org/10.1016/j.ejor.2012.05.028>.
- Archetti, C. and Bertazzi, L. (2021). Recent challenges in routing and inventory routing: E-commerce and last-mile delivery. *Networks*, 77(2):255–268. <https://doi.org/10.1002/net.21995>.
- Archetti, C., Feillet, D., and Speranza, M. G. (2015). Complexity of routing problems with release dates. *European Journal of Operational Research*, 247(3):797–803. <https://doi.org/10.1016/j.ejor.2015.06.057>.
- Arslan, A. M., Agatz, N., Kroon, L., and Zuidwijk, R. (2019). Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers. *Transportation Science*, 53(1):222–235. <https://doi.org/10.1287/trsc.2017.0803>.
- Asdecker, B. and Zirkelbach, F. (2020). What drives the drivers? a qualitative perspective on what motivates the crowd delivery workforce. *53rd Hawaii International Conference on System Sciences*. <https://doi.org/10.24251/HICSS.2020.490>.
- Auad, R., Erera, A., and Savelsbergh, M. (2020). Capacity requirements and demand management strategies in meal delivery. *Optimization Online*. [http://www.optimization-online.org/DB\\_FILE/2020/11/8125.pdf](http://www.optimization-online.org/DB_FILE/2020/11/8125.pdf).
- Auad, R., Erera, A., and Savelsbergh, M. (2022). Dynamic courier capacity acquisition in rapid delivery systems: a deep q-learning approach. *Optimization Online*. [http://www.optimization-online.org/DB\\_FILE/2022/01/8778.pdf](http://www.optimization-online.org/DB_FILE/2022/01/8778.pdf).
- Berbeglia, G., Cordeau, J.-F., and Laporte, G. (2010). Dynamic pickup and delivery problems. *European journal of operational research*, 202(1):8–15. <https://doi.org/10.1016/j.ejor.2009.04.024>.
- Brenton, J., Bowen, S., and Elliot, S. (2019). Time to cook is a luxury many families don't have. Retrieved from <https://theconversation.com/time-to-cook-is-a-luxury-many-families-dont-have-117158>. Accessed on April 2022.
- Carlsson, J. G. and Devulapalli, R. (2012). Dividing a territory among several facilities. *INFORMS Journal on Computing*, 25(4):730–742. <https://doi.org/10.1287/ijoc.1120.0536>.
- Cheng, A. (2018). Millennials are ordering more food delivery, but are they killing the kitchen, too? Retrieved from <https://www.forbes.com/sites/andriacheng/2018/06/26/millennials-are-ordering-food-for-delivery-more-but-are-they-killing-the-kitchen-t-#247cfd8a393e>. Accessed on April 2022.

- Dayarian, I. and Savelsbergh, M. (2020). Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders. *Production and Operations Management*, 29(9):2153–2174. <https://doi.org/10.1111/poms.13219>.
- Everett, H. (2019). 19 online ordering statistics every restaurateur should know in 2019. Retrieved from <https://upserve.com/restaurant-insider/online-ordering-statistics/>. Accessed on April 2022.
- Guo, B., Liu, Y., Wang, L., Li, V. O., Lam, J. C., and Yu, Z. (2018). Task allocation in spatial crowdsourcing: Current state and future directions. *IEEE Internet of Things Journal*, 5(3):1749–1764. <https://doi.org/10.1109/JIOT.2018.2815982>.
- Hakimi, S. L. (1965). Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations research*, 13(3):462–475. <https://doi.org/10.1287/opre.13.3.462>.
- Huang, Y., Zhao, L., Powell, W. B., Tong, Y., and Ryzhov, I. O. (2019). Optimal learning for urban delivery fleet allocation. *Transportation Science*, 53(3):623–641. <https://doi.org/10.1287/trsc.2018.0861>.
- Jordan, W. C. and Graves, S. C. (1995). Principles on the benefits of manufacturing process flexibility. *Management Science*, 41(4):577–594. <https://doi.org/10.1287/mnsc.41.4.577>.
- Klapp, M. A., Erera, A. L., and Toriello, A. (2018). The dynamic dispatch waves problem for same-day delivery. *European Journal of Operational Research*, 271(2):519–534. <https://doi.org/10.1016/j.ejor.2018.05.032>.
- Ledvina, K., Qin, H., Simchi-Levi, D., and Wei, Y. (2020). A new approach for vehicle routing with stochastic demand: Combining route assignment with process flexibility. Available at SSRN 3656374. <https://dx.doi.org/10.2139/ssrn.3656374>.
- Liu, Y. (2019). An optimization-driven dynamic vehicle routing algorithm for on-demand meal delivery using drones. *Computers & Operations Research*, 111:1–20. <https://doi.org/10.1016/j.cor.2019.05.024>.
- Lyu, G., Cheung, W.-C., Chou, M. C., Teo, C.-P., Zheng, Z., and Zhong, Y. (2019). Capacity allocation in flexible production networks: Theory and applications. *Management Science*, 65(11):5091–5109. <https://doi.org/10.1287/mnsc.2018.3169>.
- Morgan Stanley Research (2017). Is online food delivery about to get “amazoned?”. Retrieved from <https://www.morganstanley.com/ideas/online-food-delivery-market-expands/>. Accessed April 2022.
- Pillac, V., Gendreau, M., Guéret, C., and Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11. <https://doi.org/10.1016/j.ejor.2012.08.015>.

- Psaraftis, H. N., Wen, M., and Kontovas, C. A. (2016). Dynamic vehicle routing problems: Three decades and counting. *Networks*, 67(1):3–31. <https://doi.org/10.1002/net.21628>.
- Reyes, D., Erera, A., and Savelsbergh, M. (2018a). Complexity of routing problems with release dates and deadlines. *European Journal of Operational Research*, 266(1):29–34. <https://doi.org/10.1016/j.ejor.2017.09.020>.
- Reyes, D., Erera, A., Savelsbergh, M., Sahasrabudhe, S., and O’Neil, R. (2018b). The meal delivery routing problem. *Optimization Online*. [http://www.optimization-online.org/DB\\_FILE/2018/04/6571.pdf](http://www.optimization-online.org/DB_FILE/2018/04/6571.pdf).
- Savelsbergh, M. and Van Woensel, T. (2016). 50th anniversary invited article—city logistics: Challenges and opportunities. *Transportation Science*, 50(2):579–590. <https://doi.org/10.1287/trsc.2016.0675>.
- Shead, S. (2020). Covid has accelerated the adoption of online food delivery by 2 to 3 years, deliveroo ceo says. Retrieved from <https://www.cnbc.com/2020/12/03/deliveroo-ceo-says-covid-has-accelerated-adoption-of-takeaway-apps.html>. Accessed April 2022.
- Simchi-Levi, D. and Wei, Y. (2012). Understanding the performance of the long chain and sparse designs in process flexibility. *Operations research*, 60(5):1125–1141. <https://doi.org/10.1287/opre.1120.1081>.
- Steingoltz, M. and Picciola, M. (2019). Meals on wheels: The digital ordering and delivery restaurant revolution. Retrieved from <https://www.lek.com/insights/ei/digital-restaurant-delivery>. Accessed on April 2022.
- Ulmer, M., Thomas, B., and Mattfeld, D. (2019). Preemptive depot returns for dynamic same-day delivery. *EURO Journal on Transportation and Logistics*, 8(4):327–361. <https://doi.org/10.1007/s13676-018-0124-0>.
- Ulmer, M. W., Thomas, B. W., Campbell, A. M., and Woyak, N. (2021). The restaurant meal delivery problem: dynamic pickup and delivery with deadlines and random ready times. *Transportation Science*, 55(1):75–100. <https://doi.org/10.1287/trsc.2020.1000>.
- Union Bank of Switzerland (2018). Is the kitchen dead? Retrieved from <https://www.ubs.com/global/en/investment-bank/in-focus/2018/dead-kitchen.html>. Accessed on April 2022.
- Wang, X., Agatz, N., and Erera, A. (2018). Stable matching for dynamic ride-sharing systems. *Transportation Science*, 52(4):850–867. <https://doi.org/10.1287/trsc.2017.0768>.
- Yeo, L. (2021). Which company is winning the restaurant food delivery war? Retrieved from <https://secondmeasure.com/datapoints/food-delivery-services-grubhub-uber-eats-doordash-postmates>. Accessed April 2022.
- Yildiz, B. and Savelsbergh, M. (2019a). Provably high-quality solutions for the meal delivery routing problem. *Transportation Science*, 53(5):1372–1388. <https://doi.org/10.1287/trsc.2018.0887>.

- Yildiz, B. and Savelsbergh, M. (2019b). Service and capacity planning in crowd-sourced delivery. *Transportation Research Part C: Emerging Technologies*, 100:177–199. <https://doi.org/10.1016/j.trc.2019.01.021>.
- Zhong, H., Hall, R. W., and Dessouky, M. (2007). Territory planning and vehicle dispatching with driver learning. *Transportation Science*, 41(1):74–89. <https://doi.org/10.1287/trsc.1060.0167>.

# Appendices

## A Construction of base courier regions

This section briefly describes the construction procedure for base courier regions. The goal of this step is to design more compact courier operating areas so couriers may have specific knowledge of where they will spend their block, ultimately improving their satisfaction. At the core of this procedure, the sets of pickup locations defining each of the courier regions are determined by solving a  $p$ -median optimization model (Hakimi, 1965). We use a travel time-based dissimilarity as the objective to be minimized by the model, to facilitate compactness of the resulting regions.

Mathematically, given a set of pickup locations  $P$ , let  $S(p, p')$  be pairwise dissimilarity between pickup locations  $p \in P$  and  $p' \in P$ , and let  $m \geq 1$  be the number of regions to construct. This step determines a partition  $\{P_{r_i}\}_{i=1}^m$  of the set of pickup locations  $P$  and a centroid for each subset of pickup locations  $P_{r_i}$  corresponding to one of its elements, such that the total sum of pairwise dissimilarities between the centroids and the pickup locations assigned to each of them is minimized. To formulate the corresponding optimization problem, consider the following decision variables:

$$v_{p,p'} = \begin{cases} 1 & \text{if pickup location } p \in P \text{ is assigned to base courier region with centroid } p' \in P \\ 0 & \text{otherwise} \end{cases}$$

To construct the base courier regions, we solve the following integer program:

$$\min \sum_{p \in P} \sum_{p' \in P} S(p, p') v_{p,p'} \quad (4a)$$

$$\text{s.t.} \quad \sum_{p' \in P} v_{p,p'} = 1, \quad \forall p \in P \quad (4b)$$

$$\sum_{p \in P} v_{p,p} = m \quad (4c)$$

$$v_{p',p} \leq v_{p,p}, \quad \forall p \in P, \quad \forall p' \in P \quad (4d)$$

$$v_{p,p'} \in \{0, 1\}, \quad \forall p \in P, \quad \forall p' \in P \quad (4e)$$

Objective (4a) minimizes the total sum of pairwise dissimilarities between the each of the  $m$  centroids and the elements allocated to the corresponding region. Constraint set (4b) requires that every pickup location is assigned to one of the  $m$  centroids. Constraint (4c) enforces the selection of  $m$  centroids from the set of pickup locations (each defining a different base courier region). Lastly, Constraint set (4d) limits the assignment of pickup locations to the selected centroids.

Given an optimal vector  $v^*$ , let  $\{p_1, \dots, p_m\}$  be the set of selected region centroids, *i.e.*, be the set of pickup locations  $p$  satisfying  $v_{p,p}^* = 1$ . Then for  $i \in \{1, \dots, m\}$ , we initialize base courier region  $r_i$  by setting  $P_{r_i} = \{p \in P : v_{p,p_i}^* = 1\}$  as its set of pickup locations.

To account for the separation distance between pickup locations as well as order volume in the construction of base courier regions, we use  $S(p, p') \doteq o_p \cdot \tau_{\ell_p, \ell_{p'}} \cdot \tau_{\ell_{p'}, \ell_p}$ ,  $\forall p, p' \in P$ , where  $o_p$  is the number of orders placed to pickup

location  $p \in P$  during  $T$  (in practice,  $o_p$  can be estimated using historical order volumes at the different pickup locations).