Fleet Sizing and Service Region Partitioning for Same-Day Delivery Systems

Dipayan Banerjee Alan L. Erera Alejandro Toriello

H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology, Atlanta, Georgia 30332

September 21, 2021

Abstract

We study the linked tactical design problems of fleet sizing and partitioning a service region into vehicle routing zones for same-day delivery (SDD) systems. Existing SDD studies focus primarily on operational dispatch problems and do not consider system design questions. Prior work on SDD system design has not considered the fleet sizing decision when a service region may be partitioned into zones dedicated to individual vehicles; such designs have been shown to improve system efficiency in related vehicle routing settings. Using continuous approximations to capture average-case operational behavior, we consider first the problem of independently maximizing the area of a single-vehicle delivery zone. We characterize area-maximizing dispatching policies and leverage these results to develop a procedure for calculating optimal areas as a function of a zone's distance from the depot, given a maximum number of daily dispatches per vehicle. We then demonstrate how to derive fleet sizes from optimal area functions and propose an associated Voronoi approach to partition the service region into single-vehicle zones. We test the fleet sizing and partitioning approach in a computational study that considers two different service regions and demonstrate its pragmatism and effectiveness via an operational simulation. Using minimal computation, the approach specifies fleet sizes and builds vehicle delivery zones that meet operational requirements, verified by simulation results.

1. Introduction

Online purchasing continues to be a key factor in the growth of retail sales: total first-quarter e-commerce spending in the United States in 2020 grew by $18 billion from the previous year, an increase of 14.5%

(United States Census Bureau 2020). Driven in part by restrictions and safety concerns during the COVID-19 pandemic, U.S. online spending in May 2020 alone grew 77.8% from the previous May (Adobe Analytics 2020). Increased competition among e-commerce retailers has led to faster delivery time guarantees through next-day delivery and same-day delivery (SDD) services. Amazon offers SDD to its customers in over 8,000 cities and towns in the U.S. (Amazon 2018), and traditional retail stores such as Target and Walmart have also begun offering SDD services in recent years (Thomas 2019).

Due to low order volumes and large numbers of potential delivery locations, last-mile delivery is generally cost-inefficient in contrast to other parts of a freight logistics system. SDD presents additional challenges due to significantly tighter time constraints, in contrast to traditional last-mile delivery systems that allow for longer response times. In SDD systems, request arrival, order picking and processing, vehicle loading, and delivery all occur within the span of a service day. This contrast is evident even between SDD and next-day delivery, where most order processing occurs prior to the delivery operations that take place during the business day (Klapp et al. 2020). The result is that SDD systems often dispatch an individual vehicle on multiple routes during the service day, serving fewer customers per time unit. Controlling costs in such routing systems is of critical importance, and careful planning is required.

In recent years, the operations research community has devoted considerable attention to optimizing operational SDD problems (*i.e.*, focusing on day-to-day issues, such as dispatching and routing) (*e.g.*, Côté et al. 2019, Dayarian et al. 2020, Klapp et al. 2018a, Voccia et al. 2019). Defining aspects of such problems include the accumulation of orders over the course of a service day, a "cutoff" time after which orders are no longer eligible for SDD, and delivery deadlines that may be order-specific, as in food delivery (Reyes et al. 2018), or common to all orders placed during the same day. Decisions such as vehicle dispatch times and order acceptance and rejection are made dynamically, and are planned using heuristic methods due to the problems' underlying complexity. Operational planning problems seek to optimize metrics of efficiency, such as orders served or total routing time, while assuming fixed system features and parameters.

On the other hand, research on tactical SDD problems, focusing on design decisions made less frequently, has been more scarce (Stroh et al. 2021). Tactical decisions include the sizing or composition of delivery vehicle fleets and the selection of order cutoff times and delivery deadlines (but exclude long-term strategic decisions such as fulfillment center placement). Stroh et al. (2021) analyze tactical design issues for relatively small, fixed service regions served either by a single vehicle or a small fleet, where each vehicle serves customer locations scattered throughout the entire service region. Nevertheless, in many SDD

settings, a single depot may serve a significantly larger service region with many hundreds or even thousands of deliveries per day (perhaps even an entire metropolitan area). In such cases it is impractical to have every vehicle serving all possible customer locations, and it is much more efficient to geographically partition customers into zones with dedicated delivery vehicles (Bramel and Simchi-Levi 1995). Our motivation is precisely the design of SDD systems at this larger scale.

1.1 Contributions

Our work addresses the tactical problems of fleet sizing and region partitioning for SDD systems when each vehicle is responsible for independently serving orders within a distinct service sub-region or *zone*. We rely on analytical continuous approximation methods to ensure transparency and interpretability. Specifically:

1. Using continuous approximations to model average-case system behavior, we consider the travel of a single vehicle making same-day deliveries within a single sub-region (zone) from a single depot. Orders accumulate until a fixed cutoff time and share a common deadline, while vehicles must return to the depot by the end of the service day after delivering all orders. We consider a problem of maximizing the area of such a zone such that all same-day orders can be feasibly served with a single vehicle and characterize the structure of area-maximizing vehicle dispatching policies. We leverage this structure to propose a straightforward exact method for calculating maximal zone areas.

2. We use zone area maximization results to guide fleet sizing calculations for same-day delivery systems. Defining each vehicle's load in relation to the maximum area of its associated zone, we demonstrate how computationally inexpensive Voronoi partitioning procedures can be adapted to the SDD context to design zones for each vehicle in a fleet.

3. We validate our design approach by simulating discrete order arrivals served by an operational policy that mimics the structure of the tactical area-maximizing policies. We demonstrate that our designs perform well empirically, that they accurately predict various system metrics (such as expected number of orders served), and that they are robust to uncertainties (*e.g.* order arrivals and locations) important in practice.

The remainder of the paper is organized as follows. We briefly review the relevant literature in Section 2. In Section 3, we propose and analyze a continuous approximation model for fleet sizing in SDD systems. In Section 4, we describe a method for translating our fleet sizing results into service region partitions.

Section 5 details the computational validation of our models. Section 6 provides concluding remarks. The appendices contain technical details, proofs, and experimental data omitted from the main body.

2. Literature Review

SDD problems are closely related to the vehicle routing problem (VRP) and its variants. VRPs with probabilistic customers (*e.g.*, Ulmer et al. 2018, 2019a) provide useful models for the random nature of orders placed within a SDD system; Oyola et al. (2017, 2018) survey models and solution methods for the broader class of stochastic VRPs. The pickup and delivery VRP (Savelsbergh and Sol 1995) and the VRP with customer release times (Cattaruzza et al. 2016a) reflect the necessity for same-day orders to be picked up from a depot after the orders are placed. SDD vehicles often leave the depot multiple times over the course of a service day, as in multi-trip VRPs (Cattaruzza et al. 2016b, Paradiso et al. 2020). Operational SDD problems in the literature generally incorporate aspects of some or all of the aforementioned VRP variants as well as general characteristics of dynamic VRPs (Pillac et al. 2013, Psaraftis et al. 2016).

An example of such an operational problem is the SDD problem for online purchases (SDDP-OP). As introduced by Voccia et al. (2019), the setting of the SDDP-OP involves a fixed fleet and a single depot. The objective is to maximize the expected total number of fulfilled orders, and the problem is solved via approximate dynamic programming (DP). Côté et al. (2019) study an extension to the SDDP-OP in which vehicles are allowed to pre-emptively return to the depot. Another operational SDD model is the dynamic dispatch waves problem (DDWP) as studied in Klapp et al. (2018a,b, 2020). In the DDWP, orders arrive stochastically over the course of a service day and are served by a single uncapacitated vehicle. The objective is to minimize the sum of routing costs and penalties for unserved orders. Klapp et al. (2018b) consider the DDWP in one dimension; the deterministic version of the problem is solved via DP, and heuristics are proposed for the stochastic version. The two-dimensional DDWP is studied in Klapp et al. (2018a). In Klapp et al. (2020), a variant is studied in which orders are immediately accepted or rejected upon request.

Additional operational SDD problems studied in the literature consider drones (Dayarian et al. 2020, Ulmer and Thomas 2018), autonomous vehicles (Ulmer and Streng 2019), and other extensions (Ulmer 2017, Ulmer et al. 2019b, Yao et al. 2019). Common among these problems, as well as the SDDP-OP and DDWP, are fixed system features at the tactical level: design parameters such as fleet size and order cutoff times are assumed to be given. While it is possible to gain tactical managerial insights by repeatedly solving operational problems (see the simulation study of delivery deadlines by Ulmer (2017)), doing so

can be computationally expensive. Additionally, operational problems are often solved heuristically without optimality guarantees, potentially decreasing the reliability of any associated insights. Stroh et al. (2021), however, directly study SDD system design at the tactical level with the use of continuous approximation methods, which we briefly discuss next.

The use of continuous approximations in routing relies on the Beardwood-Halton-Hammersley (BHH) Theorem (Beardwood et al. 1959) for the traveling salesperson problem (TSP): the length of the optimal TSP tour through n points selected from a continuous distribution over a two-dimensional zone of area A approaches $\beta\sqrt{An}$ as n increases, where β is a constant. The theorem motivates a closed-form approximation of routing times as a function of the number of stops on the route. Subsequent work studies BHH-type approximations for zones of different shapes (Daganzo 1984) and with different underlying metrics (Newell and Daganzo 1986a,b). Empirical estimates of the BHH routing constant for varying tour sizes are calculated by Applegate et al. (2007) and Johnson et al. (1996) for Euclidean and rectilinear (Manhattan) distances, respectively, and Merchán and Winkenbach (2019) empirically validate continuous approximation approaches in the context of urban route distances. Examples of recent work in urban last-mile logistics that use continuous approximations for routing times include Bergmann et al. (2020), Carlsson and Song (2018), and van Heeswijk et al. (2019). Of particular relevance to our work are Lei et al. (2012) and Franceschetti et al. (2017a), in which continuous approximations are used in conjunction with discrete optimization methods to solve fleet sizing and partitioning problems for VRPs. However, these studies assume that vehicles dispatch once per day, and all customer locations are realized prior to the vehicles' departure. In contrast, to capture SDD system features, our work allows vehicles to dispatch multiple times per day and assumes delivery requests arrive dynamically over the course of the service day. Comprehensive surveys of logistical applications of continuous approximation methods are presented in Ansari et al. (2018) and Franceschetti et al. (2017b).

In Stroh et al. (2021), orders accumulate continuously over a service region at a fixed homogeneous rate per area and per time until a cutoff time. The time required to serve q orders, where q is possibly fractional, is represented by a concave, increasing dispatch time function, such as the sum of a BHH square root term and a per-order service time. Continuous approximations of both order accumulation and routing time are used to model average-case system behavior. Vehicles dispatch from a single depot, and each vehicle is responsible for orders in the entire region. All orders must be served by the end of the service day, and all vehicles must return to the depot by the end of the service day. The objective is to minimize total dispatch

time. Tactical dispatching solutions with optimality guarantees are developed for both single-vehicle fleets and infinite fleets. While the setting of Stroh et al. (2021) is similar to ours, the problems considered in this work are significantly different. First, the prior work assumes that every vehicle may serve orders from the entire region (*i.e.*, no partitioning of the service region), which is inefficient when regions grow larger. Second, Stroh et al. (2021) seek to minimize the total dispatch time under this assumption. In contrast, we explicitly seek to minimize the number of vehicles required to serve the region when it is partitioned into individual vehicle routing *zones*, as is common in practice. Finally, the prior work assumes that the order rate is constant over time. We relax this assumption in Section 3.4.

In many practical routing contexts, when the number of potential customer locations is large, it is often preferable to partition a service region into smaller zones. Our work is focused specifically on partitioning a region into single-vehicle delivery zones, where each vehicle is independently responsible for serving demand within a zone. As discussed by Zhong et al. (2007), such a scheme improves efficiency by increasing drivers' familiarity with a smaller geographical area. Given a set of "generator" points in a plane region, a *Voronoi partition* or *tessellation* of the region can be used to create zones by assigning to each generator r the set of all points that are closer to r than to any other generator. A common extension is the multiplicatively-weighted Voronoi tessellation, in which distances to each generator are multiplied by differing positive weights in order to balance the area or load of each zone. Weighted Voronoi partitions are often used in conjunction with continuous approximation methods, including for facility location (*e.g.*, Carlsson and Jia 2013, Ouyang and Daganzo 2006) and the design of vehicle routing zones (Carlsson and Delage 2013, Ouyang 2007). A centroidal Voronoi tessellation is another extension in which the generator and centroid of each zone coincide (Du et al. 1999). We propose an approximate two-stage weighted centroidal Voronoi tessellation (WCVT) scheme for service region partitioning adapted from Galvão et al. (2006), Ouyang (2007), and Ouyang and Daganzo (2006), which we discuss further in Section 4. WCVT schemes tend to produce partitions with compact, connected zones and equitable workloads without the need for solving computationally expensive integer programs. Other partitioning schemes for vehicle routing zones include those that incorporate large neighborhood search heuristics (Lei et al. 2012) and iterative ham-sandwich cuts (Carlsson 2012); however, these are less suitable for our specific problem setting. The former requires a set of deterministic customers — significantly more numerous than than the number of zones — along with a set of stochastic customers, while our problem assumes that all customers and their locations are stochastic. The latter requires, as inputs, fixed seed points corresponding to each zone, which our problem setting does

not include.

3. Area Maximization and Fleet Sizing

We consider an SDD system operating on a daily basis in a simply connected, two-dimensional service region R with a single depot from which a fleet of homogeneous, uncapacitated vehicles are dispatched. SDD orders are placed by customers beginning at the start of the service day. The system should operate such that, on average, SDD orders placed by a target cutoff time are delivered by vehicles that all return to the depot by an end-of-day deadline. Each vehicle is permitted a maximum number of dispatches from the depot. Our goal is to design this system by partitioning the region into non-overlapping *zones*, each served by a single vehicle. Specifically, we seek a partition that uses a minimum number of vehicle zones while respecting the system's aforementioned constraints.

We develop a continuous approximation optimization model of the system to solve this tactical design problem. The model is formally characterized as follows:

- The discrete order arrival process is approximated by assuming that demand for same-day deliveries (*i.e.*, orders) accumulates deterministically and continuously throughout the service region at a constant, geographically homogeneous rate of λ orders per unit area per unit time. Orders accumulate from the beginning of the service day at time $t = 0$ until the order cutoff time $t = N$. We assume without loss of generality that $\lambda = 1$ and all other parameters are scaled appropriately.
- Each vehicle in the fleet is responsible for delivering orders that accumulate in a distinct zone within the region. The fleet is assumed to be homogeneous, and vehicles are assumed to travel at a homogeneous, constant speed throughout the region.
- Vehicles are dispatched from a single depot no earlier than $t = 0$ and must return no later than the end of the service day $t = T$, where $0 < N < T$. All orders that accumulate in the service region during the interval $[0, N]$ must be served by time T.
- Each vehicle may be dispatched at most $D_{\max}$ times over the course of the service day, where the parameter $D_{\max}$ is an input selected by the system designer. Individual vehicles serving different zones may operate different dispatch schedules and may perform a different number of dispatches per day. Vehicles are uncapacitated and not constrained to carry an integer number of orders.
- The time required to complete a vehicle dispatch is given by a continuous function f_ρ that depends

only on the number of orders served, the vehicle's zone's area, and the location of the vehicle's zone's centroid.

Clearly the order process and fulfillment approximation is only appropriate in a *tactical* model of the day-to-day *operational* realities of such a system. At the operational level, the order arrival process is uncertain and might be modeled as a stochastic process, *e.g.*, a homogeneous Poisson process with rate λ. With any finite fleet, it is therefore impossible to guarantee that all orders received by the cutoff time N can be served while meeting the deadline T in every operational realization. Hence, the tactical cutoff time N and deadline T are "targets" used for planning purposes and may be dynamically adjusted at the operational level, as we discuss in Section 5. As stated earlier, each vehicle also serves a distinct zone at the operational level.

To characterize the vehicle dispatch time approximation more completely, consider a single-vehicle zone with area $A \geq 0$. Let $\rho \geq 0$ represent the travel time from the depot to the zone's centroid. We model the total time a vehicle takes to serve n orders in this zone with a single dispatch using four distinct components:

(i) a constant setup time at the depot α, with $\alpha \geq 0$;

(ii) a total linehaul time 2ρ, representing the vehicle's travel time to and from the zone;

(iii) a BHH routing time between orders $\beta\sqrt{An}$, where $\beta > 0$ is an appropriately chosen BHH constant;

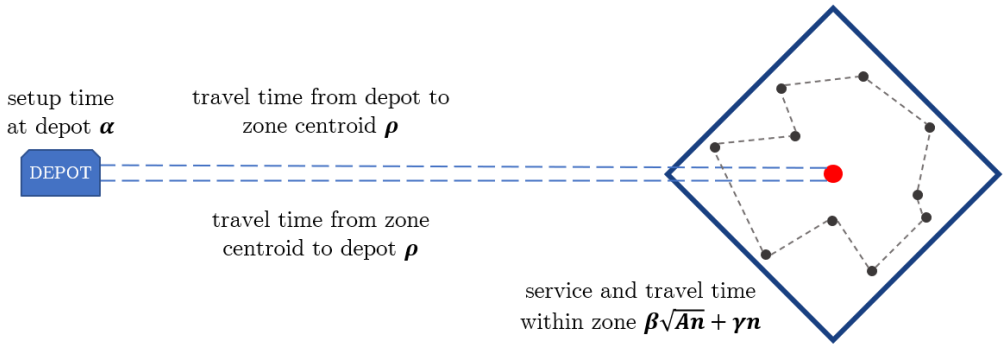
(iv) a service time γn proportional to the number of orders, where $\gamma \geq 0$ is the service time per order;

The use of the BHH function to create an approximation of travel time within the zone relies on the assumption that order locations are independently sampled from a continuous distribution over the zone (Stroh et al. 2021). Figure 1 illustrates the total dispatch (setup, travel, and service) time of a single vehicle performing a single dispatch. In general, this is a slightly conservative estimate, since, in an optimal tour, the vehicle would be unlikely to travel as far as the centroid of the depot to begin serving orders.

The (continuous) number of orders that accumulate in the zone over a period of length τ is $A\tau$. Therefore, the total time a vehicle takes to serve the orders accumulating over a period of length τ with a single dispatch is given by the dispatch time function $f_\rho(A,\tau) = 2\rho + \alpha + \beta A\sqrt{\tau} + \gamma A\tau$. The dispatch time function f_ρ is continuous, increasing in A for fixed $\tau > 0$, and non-negative. Additionally, for fixed $A > 0$, f_ρ is increasing and strictly concave in τ.

Finally, the assumption that vehicles are uncapacitated is justified by the small physical size of products commonly offered for SDD (office supplies, household goods, etc.), the comparatively smaller order volume

Figure 1: Components of total dispatch time



served, and by the tendency for delivery vehicles to operate under full capacity when time constraints significantly restrict flexibility (Klapp et al. 2018a,b, Stroh et al. 2021). The constant order rate assumption is relaxed in Section 3.4, and the assumption regarding the homogeneity of vehicle travel is revisited in Section 5.2.

We now seek to determine the minimum number of vehicles required to feasibly serve the region given this tactical approximation model, and to do so we solve a *zone area maximization problem*, the solution of which leads directly to a minimum fleet size via integration (Daganzo 2005, Erera 2000). Specifically, given a fixed centroid with associated linehaul travel time ρ, we seek to determine the maximum area of a zone around the centroid, independent of other zones and service region boundaries, such that a vehicle can feasibly serve all zone orders with at most $D_{\max}$ dispatches.

3.1 Zone area maximization

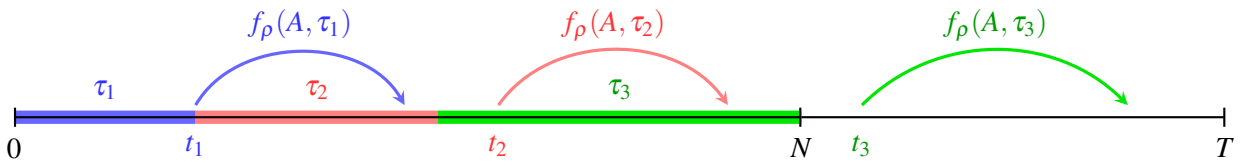
Consider a single-vehicle zone, associated dispatch time function f_ρ, number of dispatches D, and service day defined by N and T. Without loss of generality, assume $\alpha = 0$ unless stated otherwise for convenience (since we can include half of the setup time into the linehaul travel time ρ).

We assume that orders are not differentiated geographically for batching purposes; *i.e.*, when the vehicle is dispatched with n orders, those orders are distributed uniformly across the zone with a density n/A. We can therefore describe a dispatching policy with exactly D dispatches by a D-tuple of ordered pairs $\big((t_1,\tau_1),(t_2,\tau_2),\ldots,(t_D,\tau_D)\big)$. The dispatch *departure times* $t_1,\ldots,t_D$ satisfy $0 \le t_1 \le \ldots \le t_D \le T$. For each $d \in [D]$, where $[D] = \{1,2,\ldots,D\}$, the corresponding τ_d represents the duration of time during which orders accumulate to be served by the d-th dispatch. We refer to τ_d as the *accumulation time* of the d-th dispatch, and $q_d = \lambda A \tau_d = A\tau_d$ as the *quantity* of the d-th dispatch. For the policy to be feasible, it must serve all orders and therefore satisfy $\sum_{d\in[D]} \tau_d = N$. A feasible policy also cannot serve orders before they

accumulate; *i.e.*, $\sum_{d\in[D]} \tau_d \leq t_d$ for all D. Since orders placed at different times are geographically invariant, we assume without loss of generality that all orders for the d-th dispatch accumulate before any orders accumulate for the $(d+1)$-th dispatch; thus, the accumulation intervals for subsequent dispatches partition the time from 0 to N.

For example, consider a dispatching policy for a setting with $D = 3$, $N = 65$ and $T = 100$. Suppose the vehicle first departs from the depot at $t_1 = 15$ with all of the orders accumulated in the time interval $[0, 15]$, departs again at $t_2 = 40$ with all of the orders that accumulated in the time interval $[15, 35]$, and departs again with all of the remaining orders at $t_3 = 70$. The policy would be denoted as $\big((15, 15), (40, 20), (70, 30)\big)$. Figure 2 illustrates this dispatching policy to scale (with arrows representing the dispatch departure and arrival times) when $A = 1$ and $f_\rho(A, \tau) = 8 + 2A\sqrt{\tau} + 0.1A\tau$.

Figure 2: Dispatching policy example



To ensure feasibility, the area to be served by a given policy cannot be so large as to entail overlapping dispatches or a violation of the deadline T. Our goal is to find the maximum area associated with at least one dispatching policy that serves the area feasibly. While we are restricted to policies having at most $D_{\max}$ dispatches, the actual number of dispatches in an area-maximizing policy is left as a decision. For example, consider a zone centered at the depot ($\rho = 0$). In a one-dispatch policy, the vehicle is idle during the interval $[0, N]$, so introducing a second dispatch to serve some orders during this interval leads to a more "efficient" use of the vehicle. Thus, it may initially seem that using as many dispatches as possible (*i.e.*, $D_{\max}$) is desirable; this is not necessarily the case.

Define $\mathcal{A}_D(\rho)$ as the maximum area of the zone for a given ρ if the vehicle makes exactly D dispatches over the course of the service day. Also, define $\mathcal{A}_D^+(\rho) = \max\{\mathcal{A}_d(\rho) \mid d \leq D\}$, the maximum area of the zone if the vehicle can dispatch at most D times. Our goal is to characterize $\mathcal{A}_{D_{\max}}^+(\rho)$; from a tactical planning and design perspective, we are primarily interested in the cases where $D_{\max}$ is small and we can simply find $\mathcal{A}_{D_{\max}}^+(\rho)$ by calculating $\mathcal{A}_D(\rho)$ for all $D \in [D_{\max}]$.

Two conditions are necessary for the vehicle to feasibly complete D dispatches for some area $A > 0$: $2D\rho < T$ and $2\rho < T - N$. Because the minimum dispatch length is 2ρ, the former condition ensures that

D dispatches can take place in the time interval $[0,T]$. Since the last dispatch must depart no earlier than N to serve all orders, the latter condition ensures that a dispatch departure time is feasible in the time interval $[N,T]$. If these conditions are not satisfied, we define $\mathcal{A}_D(\rho) = 0$. Observe that, for a given value of ρ, it may not be feasible for the vehicle to complete $D_{\max}$ dispatches. As such, for all ρ, define $D_{\max}(\rho) \leq D_{\max}$ to be the maximum number of dispatches the vehicle can feasibly complete while respecting the dispatch constraint imposed by the system designer.

For fixed $\rho \geq 0$ and $0 < D \leq D_{\max}(\rho)$, we define $\mathcal{A}_D(\rho) \geq 0$ as the optimal value of the following non-convex optimization problem:

$$\mathcal{A}_D(\rho) = \max_{t,\tau,A} \; A \tag{1a}$$

$$\text{s.t. } t_d \geq \sum_{i=1}^{d} \tau_i \qquad \forall d \in [D], \tag{1b}$$

$$t_{d+1} \geq t_d + f_\rho(A, \tau_d) \qquad \forall d \in [D-1], \tag{1c}$$

$$T \geq t_D + f_\rho(A, \tau_D), \tag{1d}$$

$$\sum_{d=1}^{D} \tau_d = N, \tag{1e}$$

$$\tau_d \geq 0 \qquad \forall d \in [D]. \tag{1f}$$

Note that D is an input to (1). For each dispatch, constraints (1b) restrict the vehicle to depart with no more orders than have accumulated. For each dispatch d within the first $D-1$ dispatches, constraints (1c) ensure that the vehicle returns to the depot after the d-th dispatch before departing on the $(d+1)$-th dispatch. Similarly, constraint (1d) ensures that the vehicle returns to the depot for the last time no later than T. Constraint (1e) ensures that all of the orders in the interval $[0,N]$ are served, and constraints (1f) restrict order sizes to be non-negative. Additionally, constraints (1b) and (1e) together ensure that the last dispatch departs no earlier than N. Note that problem (1) always has a feasible solution with $A = 0$; thus, the necessary feasibility conditions are also sufficient for the feasibility of D dispatches when the zone area is not fixed in advance.

3.2 Optimal policy structure

When $D = 1$, if the feasibility conditions are satisfied, an optimal area-maximizing policy is clear: the sole dispatch should depart at N with all accumulated orders and return at T. This implies $\tau_1 = N$ and

$f_\rho(A,\tau_1) = T - N$, so $2\rho + \beta A\sqrt{N} + \gamma AN = T - N$. Solving the equation for A gives

$$\mathcal{A}_1(\rho) = \frac{T - N - 2\rho}{\gamma N + \beta\sqrt{N}}. \tag{2}$$

Observe that the feasibility condition $2\rho < T - N$ implies $\mathcal{A}_1(\rho) > 0$. However, problem (1) does not readily admit a closed-form optimal solution for $D > 1$. To develop a procedure for calculating $\mathcal{A}_D(\rho)$ for general D, we now state a series of results regarding the structure of area-maximizing policies.

For $d \in [D-1]$, we say that there is *waiting after* dispatch d if $t_d + f_\rho(A,\tau_d) < t_{d+1}$. Similarly, there is waiting after dispatch D if $t_D + f_\rho(A,\tau_D) < T$. For some area A, we say that an associated policy $\big((t_1,\tau_1),\ldots,(t_D,\tau_D)\big)$ *involves waiting* if there is waiting after at least one dispatch $d \in [D]$. The policy illustrated in Figure 2 involves waiting, since there is waiting after every dispatch. The following lemma states that waiting after even a single dispatch can be "redistributed" such that there is waiting after every dispatch; the proof is deferred to Appendix A.

Lemma 1. *Consider an area $A > 0$ and an associated feasible policy $P = \big((t_1,\tau_1),\ldots,(t_D,\tau_D)\big)$. If P involves waiting, there exists a feasible policy $\hat{P} = \big((\hat{t}_1,\hat{\tau}_1),\ldots,(\hat{t}_D,\hat{\tau}_D)\big)$ serving A that includes waiting after every dispatch $d \in [D]$.*

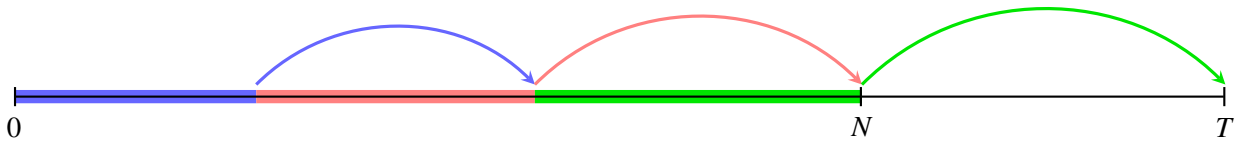
If a policy involves waiting after every dispatch, constraints (1c) and (1d) hold strictly. Thus, the area A can be increased slightly while keeping the solution feasible. The next theorem follows from this idea.

Theorem 2. *Let $D \le D_{\max}(\rho)$. Any optimal D-dispatch policy for $\mathcal{A}_D(\rho)$ does not involve waiting.*

Proof. Let $\mathcal{A}_D(\rho) > 0$ and let $P = \big((t_1,\tau_1),\ldots,(t_D,\tau_D)\big)$ be an associated optimal D-dispatch policy that involves waiting. By Lemma 1, there exists some optimal D-dispatch policy $\hat{P} = \big((\hat{t}_1,\hat{\tau}_1),\ldots,(\hat{t}_D,\hat{\tau}_D)\big)$ that includes waiting after every dispatch. By the continuity of f_ρ, there exists some $\varepsilon > 0$ such that $\hat{P}$ is feasible for an area $\mathcal{A}_D(\rho) + \varepsilon$, a contradiction. □

Therefore, when $A = \mathcal{A}_D(\rho)$, constraints (1c) and (1d) hold at equality. This result is consistent with other SDD contexts (Klapp et al. 2018b, Stroh et al. 2021), where we expect the vehicle to operate continuously until the end of the service day once it makes its first dispatch. A direct consequence of the theorem is that, at optimality, $t_1 = \tau_1$ and the last dispatch returns to the depot exactly at T. As in Lemma 1, if $t_1 > \tau_1$, the first dispatch can depart earlier to introduce waiting before the second dispatch, so the policy (and asso-

Figure 3: No-wait dispatching policy satisfying conditions (a) and (b) of Theorem 3



ciated area) cannot be optimal. The next theorem, proved in Appendix A, leads to a full characterization of policies that maximize the zone area when at most $D_{\max}$ dispatches are allowed.

Theorem 3. *Let $D \le D_{\max}(\rho)$. If $\mathcal{A}_D(\rho) = \mathcal{A}^+_{D_{\max}}(\rho)$, the optimal D-dispatch policy satisfies the following conditions:*

(a) each dispatch takes all accumulated, unserved orders at the time of departure, and

(b) the final dispatch departs at N.

Equivalently, when $A = \mathcal{A}_D(\rho) = \mathcal{A}^+_{D_{\max}}(\rho)$, constraints (1b) *hold at equality.*

Figure 3 illustrates an example of a dispatching policy that satisfies the conditions in Theorems 2 and 3. We next discuss how to find such policies and their uniqueness.

3.3 Calculating maximum zone areas

We define a new function $g_\rho(A,t)$ for all $A > 0$ and $t \ge 2\rho$ as follows: if $f_\rho(A,\tau) = t$, then $g_\rho(A,t) = \tau$. In other words, given an area A and a total dispatch duration t, $g_\rho(A,t)$ is the accumulation time of the dispatch. The function has a closed-form expression; when $f_\rho(A,\tau) = 2\rho + \beta A\sqrt{\tau}$,

$$g_\rho(A,t) = \left(\frac{t-2\rho}{\beta A}\right)^2. \tag{3a}$$

When $f_\rho(A,\tau) = 2\rho + \beta A\sqrt{\tau} + \gamma A\tau$ and $\gamma > 0$,

$$g_\rho(A,t) = \left(\frac{-\beta A + \sqrt{(\beta A)^2 + 4\gamma A(t-2\rho)}}{2\gamma A}\right)^2. \tag{3b}$$

A short derivation of both expressions appears in Appendix A. By definition, for a fixed t, g_ρ is monotonically decreasing in A. Additionally, for a fixed A, g_ρ is monotonically increasing in t.

Define $f^{(0)}_\rho(A,\tau) = \tau$ and $f^{(d)}_\rho(A,\tau) = f_\rho(A, f^{(d-1)}_\rho(A,\tau))$ for all $d \ge 1$; $f^{(d)}$ is the duration of the d-th dispatch when the first occurs at time (and accumulates) τ. Similarly, define $g^{(0)}_\rho(A,t) = t$ and $g^{(d)}_\rho(A,t) =$

$g_\rho(A, g_\rho^{(d-1)}(A,t))$ for all $d \geq 1$; here, $g^{(d)}$ is the accumulation time for the $(D-d+1)$-th dispatch when the D-th (final) dispatch duration is t. A D-dispatch policy that satisfies the conditions in Theorems 2 and 3 has $\tau_D = g_\rho^{(1)}(A, T-N)$, $\tau_{D-1} = g_\rho^{(2)}(A, T-N)$, and so on. Because the sum of all accumulation times is N, the associated area A must satisfy

$$\sum_{d=1}^{D} g_\rho^{(d)}(A, T-N) = N. \tag{4}$$

The quantity $\sum_{d=1}^{D} g_\rho^{(d)}(A, T-N)$ is continuous and monotonically decreasing in A, so (4) can have at most one solution. Indeed, when $2D\rho < T$ and $2\rho < T-N$, the area can be chosen such that $\sum_{d=1}^{D} g_\rho^{(d)}(A, T-N)$ is arbitrarily small or arbitrarily large. Thus, for a given feasible D, there is exactly one policy and associated area that satisfies the conditions in Theorems 2 and 3, which we can find by numerically solving (4).

Suppose now that $\mathcal{A}_D(\rho) = \mathcal{A}_D^+(\rho)$; the zone area is maximized for exactly D dispatches when at most D dispatches are allowed. Then, an additional consequence of these results allows us to easily verify, without solving (4), whether the area $\mathcal{A}_D^+(\rho)$ can be feasibly served with exactly $D+1$ dispatches. The proof of the following result is contained in the appendix.

Proposition 4. *Consider $D \leq D_{\max}(\rho) - 1$. Assuming $\mathcal{A}_D(\rho) = \mathcal{A}_D^+(\rho)$, then $\mathcal{A}_{D+1}(\rho) \geq \mathcal{A}_D^+(\rho)$ if and only if*

$$\sum_{d=1}^{D} f_\rho^{(d)}(\mathcal{A}_D^+(\rho), 0) \leq N. \tag{5}$$

Suppose condition (5) does not hold. Then, for any $D' > D$,

$$\sum_{d=1}^{D'} f_\rho^{(d)}(\mathcal{A}_D^+(\rho), 0) \geq \sum_{d=1}^{D} f_\rho^{(d)}(\mathcal{A}_D^+(\rho), 0) > N.$$

It follows that, if $\mathcal{A}_D^+(\rho) = \mathcal{A}_D(\rho)$ and $\mathcal{A}_{D+1}(\rho) < \mathcal{A}_D^+(\rho)$, then $\mathcal{A}_{D'}(\rho) < \mathcal{A}_D^+(\rho)$ for all $D' > D$. We may therefore conclude that, for a given ρ, the optimal areas are non-decreasing in the number of dispatches until some optimal D^*, after which $\mathcal{A}_{D^*}(\rho) > \mathcal{A}_{D'}(\rho)$ for all $D' \in \{D^*+1, \ldots, D_{\max}\}$. Thus, the problem of calculating $\mathcal{A}_{D_{\max}}^+(\rho)$ is equivalent to finding D^* and the corresponding area $\mathcal{A}_{D^*}(\rho)$. Algorithm 1 consolidates our results into an iterative root-finding procedure for solving the zone area maximization problem. For a given ρ and maximum number of dispatches $D_{\max}$, the algorithm calculates $\mathcal{A}_{D_{\max}}^+(\rho)$ by determining D^* and the corresponding area $\mathcal{A}_{D^*}(\rho)$.

The specific method used to solve $\sum_{d=1}^{D} g_\rho^{(d)}(A, T-N) = N$ for A should be chosen with parameters appropriately tuned to ensure numerical precision. While we have proven the mathematical correctness

Algorithm 1: Determining maximum zone area with at most $D_{\max}$ dispatches

Input : N, T, ρ, $D_{\max}$, $D_{\max}(\rho)$, and dispatch time function $f_\rho(A,\tau)$

Output: optimal number of dispatches D^* and corresponding area $\mathcal{A}^+_{D_{\max}}(\rho) = \mathcal{A}_{D^*}(\rho)$

initialize $\mathcal{A}_1(\rho) = \frac{T-N-2\rho}{\gamma N+\beta\sqrt{N}}$ and $D^* = 1$

for $D = 2,\ldots,D_{\max}(\rho)$ **do**

  if $\sum_{d=1}^{D-1} f_\rho^{(d)}(\mathcal{A}_{D-1}(\rho),0) \le N$ **then**

    set $\mathcal{A}_D(\rho) \leftarrow A$ s.t. $\sum_{d=1}^{D} g_\rho^{(d)}(A, T-N) = N$

    set $D^* \leftarrow D$

  else

    break

set $\mathcal{A}^+_{D_{\max}}(\rho) \leftarrow \mathcal{A}_{D^*}(\rho)$

of Algorithm 1 in theory, numerical root finding methods in practice are correct within a certain tolerance. This tolerance should be negligible relative to the range of areas under consideration. We use the `scipy.optimize.brenth` method in Python (Brent 1973, SciPy 2019) with the default absolute tolerance of 2×10^{-12}. The `scipy.optimize.brenth` method, like many other root-finding routines, requires the user to provide lower and upper bounds on the value of A that solves the equation. A suitable lower bound on this area is $\mathcal{A}_1(\rho)$; we next discuss upper bounds.

In order to obtain an absolute limit on the area a vehicle can serve under any circumstances, we consider the limiting behavior of $\mathcal{A}^+_{D_{\max}}(\rho)$ as $D_{\max}\to\infty$, and whether this limit is bounded. Consider the limit $\lim_{D\to\infty}\mathcal{A}_D(\rho)$; for all $\rho>0$, this limit is equal to zero because we violate the condition $2D\rho < T$ when D is sufficiently large. Therefore, a finite $\hat{D}$ maximizes $\mathcal{A}_D(\rho)$, and this quantity is bounded above by $\mathcal{A}^+_{\hat{D}}(0)$. When $\rho=\alpha=0$, the D-dispatch policy associated with $\mathcal{A}_D(0)$ is also feasible for $\mathcal{A}_{D+1}(0)$ if an additional dispatch with zero quantity is appended. Hence, $\mathcal{A}_1(0)\le\mathcal{A}_2(0)\le\mathcal{A}_3(0)\le\cdots$ by induction, and $A_\infty = \lim_{D\to\infty}\mathcal{A}_D(0) = \lim_{D_{\max}\to\infty}\mathcal{A}^+_{D_{\max}}(0)$ is nonzero.

It can be easily seen that A_∞ is finite. For any number of dispatches D and area A, a lower bound for the total dispatch time across all dispatches is $\gamma AN+\beta A\sqrt{N}$ by the concavity of f_ρ for fixed A. Because all dispatches must take place in the time interval $[0,T]$, it follows that $\mathcal{A}_D(\rho)\le\frac{T}{\gamma N+\beta\sqrt{N}}$ for any ρ and D. Therefore, A_∞ is bounded above by this finite quantity as well. By our previous results, A_∞ is the unique area that satisfies

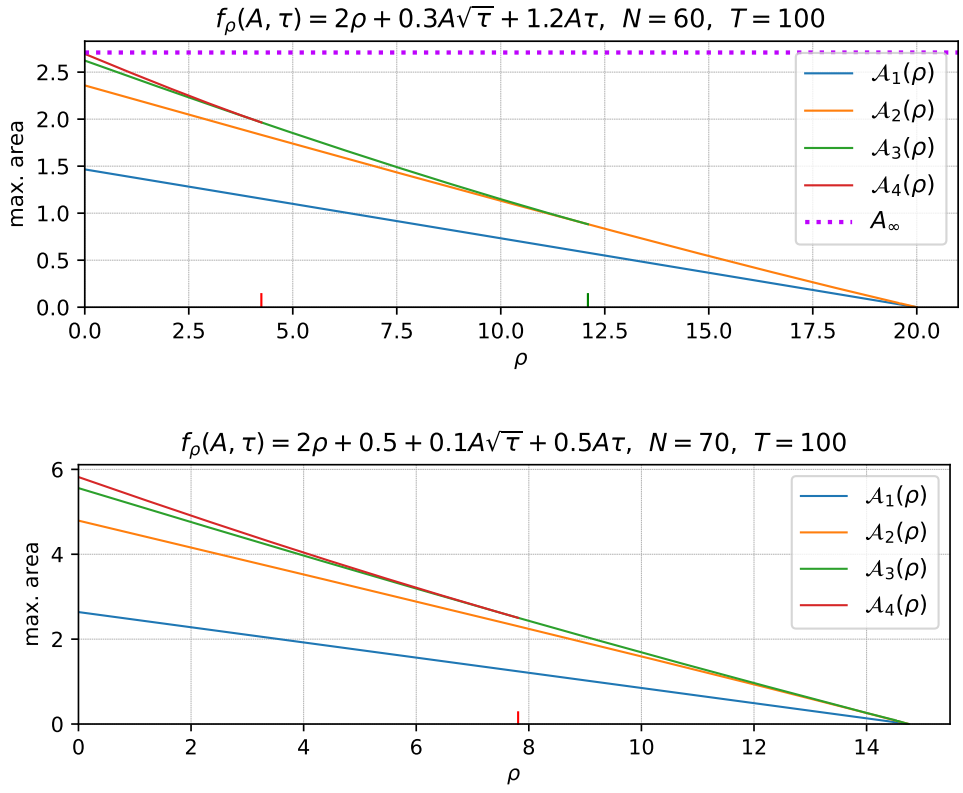
$$\sum_{d=1}^{\infty} g_0^{(d)}(A_\infty, T-N) = N. \tag{6}$$

While this series likely has no closed form (even when $\gamma = 0$), we empirically observe that the terms of the series rapidly approach zero, and A_∞ can be accurately estimated using only the first few terms of the series; in our experience, the effect of using more than eight to ten terms is negligible. Additionally, for all D, $\mathcal{A}_D(\rho)$ is bounded above by $\mathcal{A}_D(\rho - \varepsilon)$ for all $\varepsilon \in [0, \rho]$. It follows that, when $\rho > 0$, the limit $\lim_{D_{\max} \to \infty} \mathcal{A}^+_{D_{\max}}(\rho) = \mathcal{A}^+_{\hat{D}}(\rho)$ is similarly finite and bounded above by $\mathcal{A}^+_{\hat{D}}(0) < A_\infty$.

Figure 4 presents optimal area functions for two different parameter settings with $T = 100$ for all ρ such that $\mathcal{A}_1(\rho) > 0$. For each $D \leq 4$, the function $\mathcal{A}_D(\rho)$ is plotted for all ρ such that using exactly D dispatches induces a greater maximum area than using fewer than D dispatches. For each D, the smallest value of ρ for which using exactly D dispatches is no longer strictly preferable to using less than D dispatches is marked on the horizontal axis in the corresponding color. For example, in the first plot, using exactly four dispatches does not induce a greater maximum area than using three dispatches when $\rho \geq 4.25$; this cutoff is marked in red on the horizontal axis.

When $\alpha > 0$, we compute the optimal area functions by first assuming $\alpha = 0$ and calculating optimal areas as above, then shifting the functions to the left by $\frac{\alpha}{2}$. In the first case with $\alpha = 0$, we estimate A_∞ by approximating the infinite series in (6) with its first ten terms. We observe well-behaved, although

Figure 4: Optimal area function examples



not necessarily linear, area functions. The optimal area functions also exhibit diminishing returns as more dispatches are added. When the functions $\mathcal{A}_D(\rho)$ and $\mathcal{A}_{D+1}(\rho)$ intersect, they do so at the point (ρ, A) where

$$\sum_{d=1}^{D} g_\rho^{(d)}(A, T-N) = \sum_{d=1}^{D+1} g_\rho^{(d)}(A, T-N) = N.$$

We also observe a relationship between N and the optimal number of dispatches. In the first case, where $N = 60$, $\mathcal{A}_2(\rho) > \mathcal{A}_1(\rho)$ for all ρ. In the second case, where $N = 70$, $\mathcal{A}_3(\rho) > \mathcal{A}_2(\rho) > \mathcal{A}_1(\rho)$ for all ρ. We close our discussion with two related results that generalize this observation. Proofs of these results are deferred to Appendix A.

Proposition 5. *Let $2 \le D \le D_{\max}(\rho)$, and suppose $\mathcal{A}_D(\rho) = \mathcal{A}^+_{D_{\max}}(\rho)$.*

(a) If $\frac{N}{T} > \frac{D}{D+1}$, the accumulation times of the optimal D-dispatch policy are monotonically decreasing.

(b) If $\frac{N}{T} = \frac{D}{D+1}$, the accumulation times of the optimal D-dispatch policy are all equal to $\frac{N}{D}$.

(c) If $\frac{N}{T} < \frac{D}{D+1}$, the accumulation times of the optimal D-dispatch policy are monotonically increasing.

Proposition 6. *For any $D \ge 2$, $\frac{N}{T} \ge \frac{D-1}{D}$ if and only if $\mathcal{A}_D(\rho) > \mathcal{A}_{D-1}(\rho) > \cdots > \mathcal{A}_1(\rho)$ for all ρ satisfying $2\rho < T - N$.*

3.4 Time-varying order rates

Until this point, we assume that the order arrival rate is a constant $\lambda = 1$. We briefly discuss a generalization of this setting. Suppose instead that customers are more likely to place SDD orders later in the day, when the delivery time guarantee T is closer to their time of order; this applies, for example, if we assume SDD demand is driven by customers' desire for shorter turnaround times. It follows that customers are more likely to place SDD orders closer to the cutoff time N.

Mathematically, we represent this customer behavior via an *order rate function*. For all $t \in [0, N]$, the order rate per unit time per unit area within the region is given by $\lambda(t)$; the function is assumed to be non-decreasing over time, strictly positive, and integrable. In this setting, it can be shown that area-maximizing policies are structurally identical to those implied by Theorems 2 and 3.

Theorem 7. *Assume $\lambda(t)$ is non-decreasing, strictly positive, and integrable. Let $D \le D_{\max}(\rho)$. If $\mathcal{A}_D(\rho) = \mathcal{A}^+_{D_{\max}}(\rho)$, the optimal D-dispatch policy satisfies the following conditions:*

(a) the policy does not involve waiting,

(b) each dispatch takes all accumulated, unserved orders at the time of departure, and

(c) the final dispatch departs at N.

The motivation and justification for Theorem 7 mirror those in the original constant-rate setting. We defer the technical discussion of this result, including its proof, to Appendix B for brevity. Theorem 7 implies an efficient method, analogous to Algorithm 1, for calculating $\mathcal{A}^{+}_{\max}(\rho)$, also detailed in Appendix B.

3.5 Fleet sizing

We now return to the problem of determining the minimum number of vehicle zones needed to serve the service region R when each vehicle can make at most $D_{\max}$ dispatches over the course of the service day. For all points $r \in R$, let ρ_r denote the travel time from the depot to r. Because the maximum total area of a zone centered at r is $\mathcal{A}^{+}_{D_{\max}}(\rho_r)$, we intuitively require at least $1/\mathcal{A}^{+}_{D_{\max}}(\rho_r)$ vehicles to serve a sub-region of unit area centered at r. Thus, the fleet size implied by the continuous approximation model is given by the value of the following integral evaluated over the service region (Erera 2000, p. 154):

$$V_{\mathrm{CA}} = \int_{r \in R} \frac{1}{\mathcal{A}^{+}_{D_{\max}}(\rho_r)} dr. \tag{7}$$

To this point, we have assumed orders arrive continuously and deterministically in order to model average-case system behavior. In practice, however, order arrivals are stochastic and discrete. Using more than V_{CA} vehicles leads to an average zone size less than $\mathcal{A}^{+}_{D_{\max}}(\rho)$ in the resulting partition, increasing the overall robustness of the system. On the other hand, using fewer than V_{CA} vehicles leads to an average zone size greater than $\mathcal{A}^{+}_{D_{\max}}(\rho)$ and will likely require adjusting other parameters (*e.g.*, shifting the order cutoff time earlier) to meet service level requirements and customer expectations on average. The exact fleet size should be determined on a case-by-case manner, with V_{CA} as a guideline, based on the preferences of system managers and the cost of operating vehicles. We use $\lceil V_{\mathrm{CA}} \rceil$ vehicles in our computational studies.

4. Partitioning

We now discuss how to apply our results from the previous section to partition a service region R into vehicle zones, using weighted centroidal Voronoi tessellation (WCVT) schemes (Galvão et al. 2006, Ouyang 2007).

For notational convenience, define the function $\mathcal{A}(\cdot)$ for all points $r \in R$ as $\mathcal{A}(r) = \mathcal{A}^{+}_{D_{\max}}(\rho_r)$. Given

an integer number of vehicles m, where m is determined using (7) as a guideline, we are interested in constructing a balanced partition of the region into m compact zones. Analogous to similar contexts where relative load is defined with respect to restrictions on vehicle capacity or driver working time, we interpret the *load* of a zone relative to its maximum possible area. Formally, we define the *centroidal load factor* (CLF) of a zone Z as $\text{area}(Z)/\mathcal{A}(\text{centroid}(Z))$. Our goal is to approximately balance the zone CLFs across R, as measured by all CLFs falling within a user-defined interval $[\text{CLF}_{\min}, \text{CLF}_{\max}]$ (*e.g.*, $[0.91, 0.99]$).

As a result, the area of each zone Z will not correspond exactly to $\mathcal{A}(\text{centroid}(Z))$. For practical purposes, it is impossible to create m zones each with a CLF of exactly 1, because it is unlikely that $m = V_{\text{CA}} \in \mathbb{Z}$ unless a synthetic instance was constructed explicitly for that purpose. In addition, by allowing a small interval for CLFs rather than requiring an exact number, we obtain more compact zones compared to when we require an exact balance.

Voronoi-type tessellations create zones by assigning to each generator point r the set of all points that are closer to r than to any other generator. An approximate WCVT can be constructed by selecting initial generator points, then iteratively updating zones, weights, and generator points until zone centroids and generator points approximately coincide. We first describe a method for selecting initial generator points, followed by an iterative WCVT procedure for approximately balancing loads.

4.1 Initial generator locations

To select initial generator locations, we use an "automated sliding procedure" similar to that proposed in Ouyang (2007) and Ouyang and Daganzo (2006); these studies focus on balancing customer demand across zones rather than CLF. Aside from various algorithmic parameters, which we describe and motivate next, the inputs to this method are the region R, maximum area function $\mathcal{A}(r)$, and the number of zones/vehicles m. We summarize our version of the method here and defer additional examples and further discussion to the aforementioned references.

We consider m closed balls or *disks* (circles in the Euclidean metric, diamonds in the rectilinear metric) $C_1, \ldots, C_m$ located within the service region R, with corresponding centers $c_1, \ldots, c_m$, where the area of each disk is proportional to $\mathcal{A}(c_i)$. We seek to pack the disks into the region without overlap by sliding the disks based on two kinds of repulsive "forces".

The first type of force occurs when two disks overlap. We quantify the degree of overlap between two

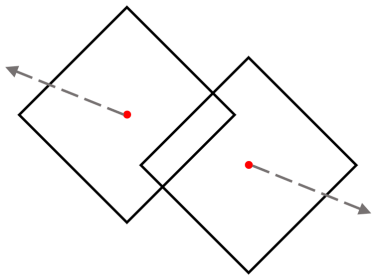
disks C_i and C_j by

$$\sigma_{ij} = 1 - \frac{\|c_i - c_j\|}{\text{radius}(C_i) + \text{radius}(C_j)}. \tag{8}$$

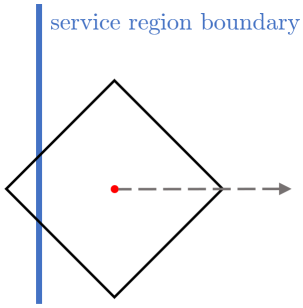
The magnitude of the repulsive force between two disks is a nonnegative increasing function $v(\sigma)$ of their degree of overlap, with $v(\sigma) = 0$ for $\sigma \leq 0$. As in Ouyang and Daganzo (2006), we take $v(\sigma)$ to be linear for $\sigma > 0$ with $\lim_{\sigma \downarrow 0} v(\sigma) > 0$. These repulsive forces act along the axis through the centers of both disks, as illustrated in Figure 5a. The second kind of force occurs when part of a disk protrudes outside of the region boundary. For a disk C_i, if $C_i \setminus R \neq \varnothing$, a repulsive force along the axis through centroid$(C_i \setminus R)$ and c_i pushes the disk back into R, as illustrated in Figure 5b. The magnitude of this boundary force, when it occurs, is always set to $(m+1) \cdot v(1)$ to ensure that overlap forces never push disks completely outside of R.

The sliding of the disks proceeds iteratively. In each iteration, forces are calculated, disks are moved a small distance based on a step-size parameter μ, and areas are updated based on the new locations of the disk centers. As recommended by Ouyang and Daganzo (2006), we decrease the step size by a multiplicative factor of Δ_μ at each iteration (*i.e.*, $\mu \leftarrow \mu \cdot \Delta_\mu$, where $0 \ll \Delta_\mu < 1$), although constant step sizes may be used as well. The algorithm terminates when the maximum overlap between any two disks is within a user-defined tolerance $\sigma_{\max}$ and no disks extend outside of the region.

Figure 5: Repulsive forces on disks for automated sliding procedure



(a) Directions of disk overlap forces



(b) Direction of boundary force

As in Ouyang and Daganzo (2006), we additionally include a shrinkage coefficient κ by which all disk areas are scaled; *i.e.*, the area of a disk at a point r is $\kappa \cdot \mathcal{A}(r)$. This ensures the eventual termination of the procedure, since disk overlaps vanish if κ is sufficiently small. If we do not find a solution in $p_{\max}$ iterations, we reduce κ by performing the update $\kappa \leftarrow \kappa \cdot \Delta_\kappa$ (where $0 \ll \Delta_\kappa < 1$), reset μ to its initial value μ_0, and continue. The initial shrinkage coefficient κ_0 should depend on the ratio of m to V_{CA}. If $m > V_{\text{CA}}$, using $\kappa_0 = 1$ is sufficient; otherwise, a value of κ_0 larger than V_{CA}/m should be used. Finally, we might also encounter instances where overlapping disks stop moving due to opposing balanced forces (such solutions are referred to as "singular points" (Okabe et al. 2000) or "degenerate locations" (Ouyang and Daganzo 2006)). We

add a small random perturbing force to each disk at each iteration to prevent this occurrence. Algorithm 2 formalizes the full method. For notational convenience, $\text{unif}(x,y)$ denotes a function which generates a value uniformly at random in the interval $[x,y]$. Additionally, $\text{norm}(\vec{x})$ denotes a function which normalizes the vector $\vec{x}$ to have unit magnitude.

Algorithm 2: Initial generator location selection

Input : $\kappa_0, \Delta_\kappa, \mu_0, \Delta_\mu, \sigma_{\max}, p_{\max}, \varepsilon_{\max}$, region R, maximum area function $\mathcal{A}(r)$, number of zones m, force function $\nu(\sigma)$

Output: generator locations $c_1, \ldots, c_m$

initialize disk centers $c_1, \ldots, c_m$ randomly within R

initialize $p = 0$, $\kappa = \kappa_0, \mu = \mu_0$, disks C_i with centers c_i and areas $\mathcal{A}(c_i)$ for all $i \in [m]$

while $(\max_{i,j}\{\sigma_{ij}\} > \sigma_{\max})$ *or* $(\bigcup_{i=1}^{m} C_i \setminus R \neq \varnothing)$ **do**

- set $p \leftarrow p+1$
- **for** disks $i \in [m]$ **do**
  - **for** $j \in [m] \setminus \{i\}$ **do**
    - set $\vec{\varphi}_{ij} \leftarrow \text{norm}\,(c_i - c_j)$ `// direction of overlap force`
  - **if** $C_i \setminus R \neq \varnothing$ **then**
    - set $\vec{\psi}_i \leftarrow \text{norm}\,(c_i - \text{centroid}(C_i \setminus R))$ `// direction of boundary force`
  - **else**
    - set $\vec{\psi}_i \leftarrow (0,0)$
  - set $\vec{F}(i) \leftarrow (m+1)\cdot\nu(1)\cdot\vec{\psi}_i + \sum_{j\neq i}\vec{\varphi}_{ij}\nu(\sigma_{ij}) + \big(\text{unif}(-\varepsilon_{\max},\varepsilon_{\max}), \text{unif}(-\varepsilon_{\max},\varepsilon_{\max})\big)$
- set $c_i \leftarrow c_i + \mu\cdot\vec{F}(i)$ for all $i \in [m]$ `// update disk centers`
- **if** $p \bmod p_{\max} = 0$ **then**
  - set $\kappa \leftarrow \Delta_\kappa \cdot \kappa$, $\mu \leftarrow \mu_0$, $p = 0$
- **else**
  - set $\mu \leftarrow \Delta_\mu \cdot \mu$
- update disk areas as $\kappa \cdot \mathcal{A}(c_i)$ for all $i \in [m]$
- update σ_{ij} for all $i, j \in [m], i \neq j$ via equation (8)

The resulting configuration of disk centers exhibit spacing between each other and with the boundaries of the region. The final value of κ represents the factor by which all zones have been scaled down relative to their maximum area. Figure 7a in Section 5 shows an example output of this algorithm. These disk centers are used as initial generator locations for the WCVT balancing procedure, which we discuss next.

4.2 Balancing zone loads

Simply creating a standard Voronoi partition from the output of the sliding procedure does not guarantee that the CLFs of the resulting zones are within the desired interval $[\text{CLF}_{\min}, \text{CLF}_{\max}]$. We employ an iterative WCVT load balancing procedure (Galvão et al. 2006, Ouyang 2007) to refine the solution.

This procedure requires the service region R to be discretized into small disjoint "blocks" $B_1, \ldots, B_\ell$. A simple approach is to discretize the region as a uniform grid, which we do in our computational studies with 0.1 mile $\times$ 0.1 mile squares. Blocks defined in geographical surveys (*e.g.* Lassiter 2011) may also be appropriate. Finer discretizations (*i.e.*, with larger ℓ) allow for more detailed zone boundaries while increasing the computational time required. The choice of ℓ and the discretization is ultimately left to the system manager. Aside from various algorithmic parameters, which we describe and motivate next, the inputs to the procedure are these blocks $B_1, \ldots, B_\ell$ with corresponding centroids $b_1, \ldots, b_\ell$, initial generator locations $c_1, \ldots, c_m$, and the maximum area function $\mathcal{A}(r)$.

At each iteration, we create zones by assigning blocks to generators based on the distance between block centroids and generators, scaled by a quantity proportional to the zone's weight and inversely proportional to the zone's maximum area. After each iteration, weights and generator locations are updated. A zone's weight is increased if its area in the previous iteration was too large and decreased if its area in the previous iteration was too small relative to its "ideal" area — its area if every zone had equal CLF — and the area of other zones. Each generator is also moved slightly closer to its associated zone's centroid. Both updates use a constant step parameter η, where $0 < \eta \ll 1$. As the process evolves, generator locations approximately converge to their corresponding zone centroids and CLFs approximately balance across zones.

Minimally, any solution should require that zones are connected. While the WCVT zones produced in any given iteration are not guaranteed to be connected, we empirically observe that disconnected components are rare, very small compared to the primary component, and organically rapidly removed by weight and centroid changes in the balancing algorithm. We additionally use the following stopping criteria:

(i) all zone CLFs falling within the user-defined interval $[\text{CLF}_{\min}, \text{CLF}_{\max}]$, and

(ii) the maximum distance between any zone's generator and centroid being less than a user-defined δ.

The first measures balance across zones, while the second criteria signals a degree of convergence of the process. From a managerial perspective, it may additionally be useful to observe and compare intermediate solutions as the process evolves. In our two computational studies, we use condition (i) with CLF intervals of

$[0.91, 0.99]$ and $[0.89, 0.94]$, respectively. To this end, we include soft lower and upper bounds $\Theta^- \leq \text{CLF}_{\min}$ and $\Theta^+ \geq \text{CLF}_{\max}$ to speed up convergence. For a zone Z_i with generator z_i, if $\text{area}(Z_i)/\mathcal{A}(z_i) > \Theta^+$, the zone's weight is increased by θ^+, and if $\text{area}(Z_i)/\mathcal{A}(z_i) < \Theta^-$, the weight is decreased by θ^-, where $0 \leq \Theta^- < \Theta^+ \leq 1$, $0 < \theta^- \ll 1$, $0 < \theta^+ \ll 1$. We also use condition (ii) with a maximum distance of 0.1 miles in both studies. Algorithm 3 formalizes the full method.

We close our methodological discussion with an overall summary of the hierarchy of the two steps in our approach: fleet sizing followed by partitioning. With inputs N, T, α, β (which may require estimation beforehand), γ, ρ, and $D_{\max}$, our area maximization procedure calculates $\mathcal{A}^+_{D_{\max}}(\rho)$; this allows the fleet sizing procedure to output an approximate minimum number of vehicles required V_{CA}. We then characterize the maximum area at any point r in the region by the function $\mathcal{A}(r) = \mathcal{A}^+_{D_{\max}}(\rho_r)$, which further allows us to

Algorithm 3: WCVT zone CLF balancing

Input : $\text{CLF}_{\min}, \text{CLF}_{\max}, \delta, \eta, \Theta^-, \Theta^+, \theta^-, \theta^+$, number of zones m, initial generator locations $\{c_1, \ldots, c_m\}$, maximum area function $\mathcal{A}(r)$, blocks $\{B_1, \ldots, B_\ell\}$

Output: SDD zones $Z_1, \ldots, Z_m$

initialize generator location $z_i \leftarrow c_i$ and weight $w_i = 1$ for all $i \in [m]$

while ($\exists$ *disconnected zone*) *or* ($\exists i \in [m]$ *s.t.* $\text{CLF}(Z_i) \notin [\text{CLF}_{\min}, \text{CLF}_{\max}]$) *or* $(\max_i\{\|z_i - \text{centroid}(Z_i)\|\} > \delta)$ **do**

- set $Z_i \leftarrow \varnothing$ for all $i \in [m]$
- set $a_i \leftarrow \dfrac{\mathcal{A}(z_i) \cdot \text{area}(R)}{\sum_{j=1}^m \mathcal{A}(z_j)}$ for all $i \in [m]$ `// area if all zones had equal CLF`
- **for** $k \in [\ell]$ **do**
  - set $i^* \leftarrow \text{argmin}_{i \in [m]} \left\{ \dfrac{w_i}{a_i} \|z_i - b_k\| \right\}$ `// find block's closest generator point, adjusted for weight and max area`
  - update $Z_{i^*} \leftarrow Z_{i^*} \cup B_k$ `// add block to corresponding zone`
- update $z_i \leftarrow (1-\eta) z_i + \eta \cdot \text{centroid}(Z_i)$ for all $i \in [m]$ `// update generators`
- **for** $i \in [m]$ **do**
  - update $w_i \leftarrow (1-\eta) w_i + \eta \cdot \dfrac{\text{area}(Z_i)}{a_i}$ `// update weight`
  - **if** $\text{area}(Z_i)/\mathcal{A}(z_i) > \Theta^+$ **then**
    - update $w_i \leftarrow w_i + \theta^+$
  - **else if** $\text{area}(Z_i)/\mathcal{A}(z_i) < \Theta^-$ **then**
    - update $w_i \leftarrow w_i - \theta^-$

define the load at each point r as $1/\mathcal{A}(r)$. Defining load in this way permits us to adapt existing partitioning methods — of which there are many — to our setting. The specific partitioning procedure we use in this paper requires the inputs listed in Algorithms 2 and 3, including a number of desired zones m chosen using V_{CA} as a guideline; it outputs a partition of the region into m disjoint single-vehicle zones.

5. Computational Study

We empirically test our approach in two computational studies to evaluate the model's approximations and predictions. We simulate SDD operational settings in which orders arrive according to a Poisson point process, and where exact vehicle routing times are computed based on order delivery locations. In our first study designed on a synthetic instance, we consider a region with homogeneous rectilinear vehicle travel and a constant order rate over time. In our second study, we demonstrate how our approach can be adapted to a realistic case study in a real-world metropolitan area, with inhomogeneous vehicle travel time and a time-varying order rate over the course of the service day, as described in Section 3.4.

During the tactical design stage, the cutoff N and deadline T are fixed. However, due to the stochastic nature of order arrivals in practice, it is impossible to guarantee that all orders accumulated by the cutoff can be feasibly served by the deadline. Therefore, at the operational level, at least one of these parameters must be flexible. As in many recent operational SDD studies (*e.g.*, Chen et al. 2020, 2021, Klapp et al. 2018a), we consider the vehicle return deadline T to be inviolable at the operational level. We instead allow for dynamic cutoff time adjustments in our operational simulations. We consider the tactical cutoff time N to be a target (possibly representing the cutoff time advertised to customers), and compare the simulated operational cutoff times to this target. We use $\lceil V_{\mathrm{CA}} \rceil$ vehicles to create service region partitions; therefore, if our approach is valid, we expect orders to be cut off slightly later than N on average at the operational level.

Each of the two studies demonstrates a different approach for empirically deriving routing functions. We compute the fleet sizing integrals numerically in Python 3.7.3 by the method of Xiao and Gimbutas (2010), as implemented in the `quadpy` library (Schlömer 2020), on polygonal service regions triangulated with the `tripy` library (Bolgert 2019). We create the service region partitions in MATLAB R2019b, and we calculate optimal TSP solutions with an arc-based integer programming formulation implemented in Gurobi 9.0.1 via Python. In our second study, we use the open source tools VeRoViz (Peng and Murray 2020) and Openrouteservice to generate customer locations and calculate travel times on the road network.

5.1 Synthetic Experiment

We first consider a service region with a total area of 194.5 square miles. The depot is located in the southwest corner of the region (see Figure 7a). Vehicle travel is given by the rectilinear (Manhattan) metric. We assume a vehicle speed of 25 miles per hour (mph), a setup time per dispatch of five minutes, and a per-order service time of 2.5 minutes.

To develop dispatch time functions for the tactical model, we require a suitable BHH routing constant β, the expected ratio between the length of the TSP tour and the square root of the number of stops n in a region of unit area (Beardwood et al. 1959). The rectilinear BHH constant is empirically estimated as 0.8943 by Johnson et al. (1996) for large n. However, it is helpful to build an estimate more appropriate for scenarios — such as ours — where the number of locations to be visited by any tour is likely to be relatively small. Within a diamond-shaped zone, our empirical estimates of the BHH constant range from 1.1162 for $n = 15$ to 0.9841 for $n = 90$. We use the constant 1.0533 (scaled appropriately) in our dispatch functions, corresponding to our estimate for $n = 30$.

The service day extends from 8 AM to 8 PM with a target order cutoff time of 2 PM. Each vehicle is allowed at most $D_{\max} = 3$ dispatches per day. Orders arrive at a rate of two per hour per square mile. To ensure that the order arrival rate is $\lambda = 1$, as in Section 3, we re-scale all parameters to be measured in increments of 30 minutes. Figure 6 displays the resulting dispatch time function, re-scaled time parameters, and optimal area functions for up to three dispatches per vehicle. The continuous approximation fleet size guideline is $V_{\mathrm{CA}} = 17.13$ vehicles for $D_{\max} = 3$. For comparison, the fleet size guidelines are 25.64 and 17.75 for $D_{\max} = 1$ and 2, respectively. We partition the service region into $\lceil V_{\mathrm{CA}} \rceil = 18$ single-vehicle zones.

The automated sliding procedure terminated after 2,705 total iterations in just under 49 seconds of runtime with a final shrinkage coefficient of $\kappa = 0.5730$. The disk centers of the resulting output, shown in

Figure 6: Maximum area functions, first computational study

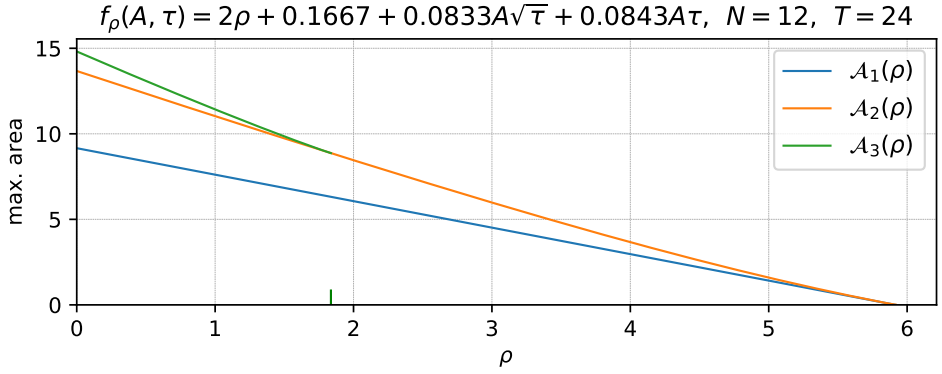
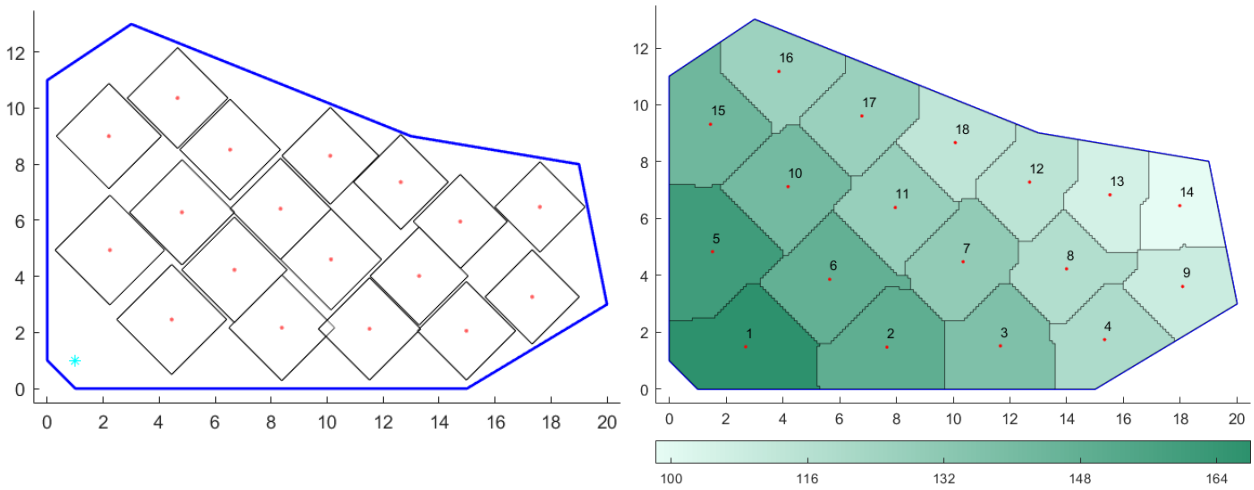


Figure 7: Service region partitioning, first computational study (axes' coordinates in miles)



(a) Output of sliding procedure

(b) Zones shaded by expected orders, 8AM to 2 PM

Figure 7a, are used as the initial WCVT generator locations. To construct the partition, the service region was discretized into 19,625 blocks via a grid comprised of 0.1 mile $\times$ 0.1 mile squares. We stopped the WCVT balancing process when every zone was connected, all CLFs fell within the range $[0.91, 0.99]$, and the largest distance between any zone's generator and centroid was below 0.1 miles. The process terminated after 50 iterations in 6 minutes, 32 seconds of runtime. Table C2 in Appendix C records the evolution of the WCVT process, and Table C1 in Appendix C lists the input parameters used in both phases of the partitioning procedure.

Note that the final average centroidal load factor of 0.9483 nearly coincides with the ratio between the minimum suggested fleet size and the actual fleet size used, $V_{\mathrm{CA}}/18 \approx 0.9516$. Individual zone areas range from 8.18 square miles to 14 square miles (corresponding to zones 14 and 1, respectively). Figure 7b illustrates the final partition with zones shaded by their expected number of orders placed by 2 PM (*i.e.*, the zone's area multiplied by N). In order to minimize expected total dispatch cost, we determine the number of dispatches for each vehicle to be the minimum D such that the associated zone area is at least $\mathcal{A}_D(\rho)$. In this case, the maximum area functions imply that zones farther from the depot require fewer dispatches to serve their area. The vehicles serving zones 1, 2, 3, 5, 6, 10, and 15 must dispatch three times each (*i.e.*, the zones' areas exceed $\mathcal{A}_2(\rho)$). The remaining zones only require dispatching twice daily.

5.1.1 Simulation results

To study the performance of the tactical partition at the operational level, we now assume that orders arrive at the same rate but as a uniform Poisson point process over the service region. Our operational policy for

a single-vehicle zone, which mimics the structure of the tactical area-maximizing policies discussed earlier, is defined as follows.

Consider the perspective of a dispatcher at the start of the service day; specifically, suppose that the zone's vehicle is designated to dispatch three times daily and serves an area A. Denote the set of accumulated unserved orders as I; at $t = 0$, $I = \varnothing$. Also, let $\text{TSP}(I \cup \{0\})$ denote the duration of a tour through the depot and all points in I, including setup and per-order service times. As the day progresses, order requests arrive and I grows. Each time an order arrives, the dispatcher re-calculates $\text{TSP}(I \cup \{0\})$. Using the zone's routing function to estimate future tour durations, the dispatcher first dispatches the vehicle to serve I at a time χ_1 so that the vehicle's expected final return to the depot occurs at the deadline. Formally, the dispatcher solves $\chi_1 + \text{TSP}(I \cup \{0\}) + f_\rho\big(A, \text{TSP}(I \cup \{0\})\big) + f_\rho^{(2)}\big(A, \text{TSP}(I \cup \{0\})\big) = T$ for χ_1. Occasionally, an order request may arrive at a time t such that $t + \text{TSP}(I \cup \{0\}) + f_\rho\big(A, \text{TSP}(I \cup \{0\})\big) + f_\rho^{(2)}\big(A, \text{TSP}(I \cup \{0\})\big) > T$. In such a case, the vehicle is dispatched immediately at $\chi_1 = t$ without serving this newest request on the first dispatch. The orders served by the first dispatch (all except potentially the last arriving) are removed from I.

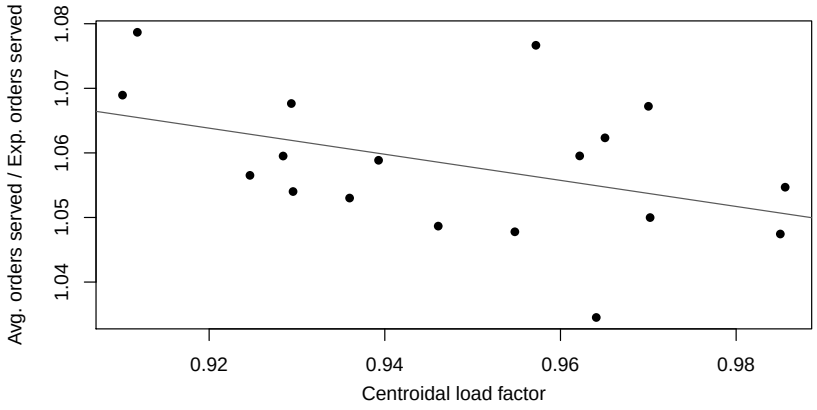
A similar process is repeated by the dispatcher to calculate the dispatch time χ_2 for the second dispatch. The vehicle obviously cannot depart on its second dispatch prior to returning to the depot from its first dispatch at some time χ_1^+. Thus, at the time of the vehicle's return, if $\chi_2 + \text{TSP}(I \cup \{0\}) + f_\rho\big(A, \text{TSP}(I \cup \{0\})\big) \geq T$, the vehicle departs on its second dispatch immediately at χ_1^+ with all orders in I. Otherwise, the dispatcher again re-calculates $\text{TSP}(I \cup \{0\})$ whenever more arrive, and instead solves $\chi_2 + \text{TSP}(I \cup \{0\}) + f_\rho\big(A, \text{TSP}(I \cup \{0\})\big) = T$ to determine the second order time. As before, an order which would push the expected final return time past T is left behind.

An analogous process is followed by the dispatcher to determine χ_3, solving the equation $\chi_3 + \text{TSP}(I \cup \{0\}) = T$ after the second dispatch's return at χ_2^+ if required, with a slight modification. For the final dispatch, if an order arrives that would require the vehicle to return after the 8 PM deadline, the order is not offered same-day delivery service and the vehicle departs immediately if it is at the depot. The time of this service rejection (or the time of the vehicle's final dispatch, if no such rejection occurs) represents the dynamically adjusted operational order cutoff time for the zone, which may occur before or after the tactical target cutoff time of 2 PM. The operational cutoff time necessarily varies across zones and from day to day, reflecting the operational necessities of a system with a inviolable end-of-day deadline T. An analogous operational policy is used for the two-dispatch zones.

We simulate a total of 100 service day realizations using this operational policy, and we record the

number of orders served and the cutoff time for each zone and each day. We summarize key results here and include additional data in Appendix C. The expected number of daily orders placed within the service region between 8 AM and 2 PM is approximately 2,334. In our simulations, the operational policy serves about 2,469 orders per day on average, an improvement of approximately 5.77% relative to the expected quantity. For each zone, Figure 8 displays the ratio between the daily average quantity served and the expected quantity served versus the zone's CLF. The value of this ratio is greater than one for each zone, indicating that each vehicle serves at least its expected number of orders on average. A downward trend in the value of this ratio is evident as the zone CLFs increase.

Figure 8: Ratios between average quantity served and expected quantity served, first computational study

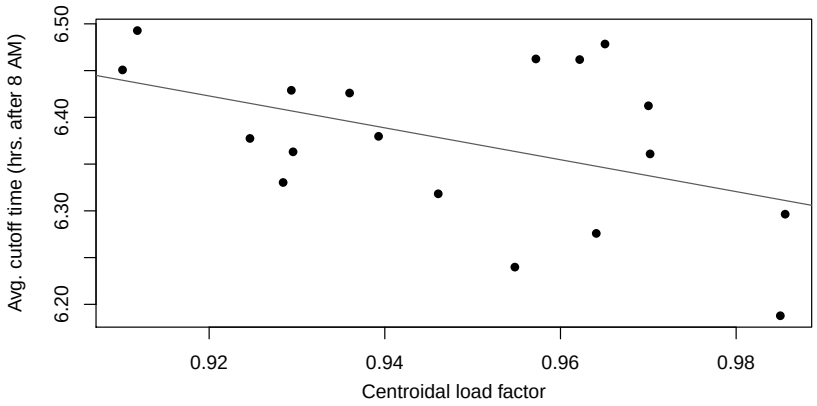


The cutoff time used for tactical planning is 2 PM. In our simulations, the mean operational cutoff time was 2:22 PM across zones. Each zone cuts off orders at or later than 2 PM an average of 86.56 days (of the 100 service days simulated). Figure 9 plots the average cutoff time by zone versus CLF; a downward trend is once again evident, and each zone cuts off orders later than 2:11 PM on average. These results suggest that our efficient tactical fleet sizing and partitioning approach produces quality solutions that exceed cutoff time targets on average. Additional data is included in Appendix C.

5.1.2 *A posteriori* operational bound

We have shown, using a fairly simple operational dispatching policy, that our tactical design scheme performs well in practice. While not a focus of this work, more sophisticated operational vehicle dispatching policies may provide improved results in terms of routing cost or orders served. As we discuss in Section 2, many recent studies focus on operational dispatching policies in the SDD context under various assumptions (*e.g.*, Côté et al. 2019, Klapp et al. 2018a, Ulmer et al. 2019b, Voccia et al. 2019). We next briefly consider the question of benchmarking our operational policy against potential improved policies.

Figure 9: Average order cutoff times, first computational study



Recall that, like our tactical policies, our operational policy does not group orders geographically: at every dispatch departure time, the vehicle departs with all accumulated unserved orders regardless of each order's delivery location within the zone. To benchmark our operational policy, we compute a "hindsight-optimal" upper bound on the number of orders served within each zone. For each daily realization and each zone, we assume the dispatcher knows in advance the time each potential order is placed as well as the order's delivery location. With this information, we seek to maximize the number of orders served over the course of the service day with a fixed number of dispatches (as prescribed by our tactical model). To reflect the notion of an order cutoff time, we include an additional constraint: if an order is served, all earlier orders within the zone must also be served.

For every daily realization, we solve the hindsight-optimal problem for zones 1, 3 (the largest and smallest three-dispatch zones, respectively), 11, and 14 (the largest and smallest two-dispatch zones, respectively). We use the best upper bound on the objective after 30 minutes as an *a posteriori* bound on the quantity of orders served. We include our MIP formulation of this *a posteriori* problem with additional computational details in Appendix C. For each zone, Table 1 reports the average quantity served by our operational policy, the average objective value of the best integer-feasible solution found during the optimization, and the average best upper bound on the objective. We also report the average and standard deviation across realizations of the gap between the operational policy and upper bound as well as the best solution and upper bound (*i.e.*, the MIP optimality gap).

Within the family of operational policies that use fixed zone boundaries and a fixed number of dispatches for each vehicle, these results show that our policy performs well with respect to the daily number of orders served. For the two-dispatch zones, our operational policy is often optimal, and the average upper bounds

Table 1: *A posteriori* upper bounds

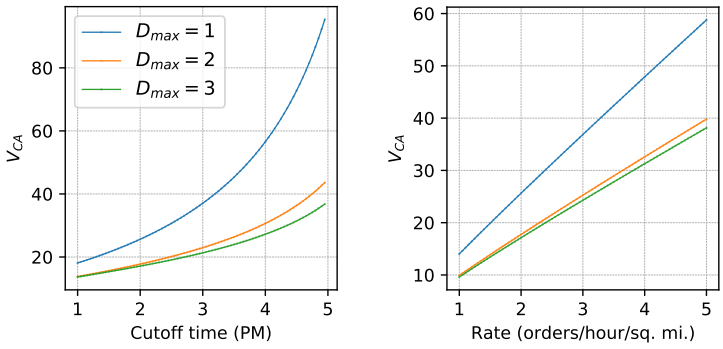
Zone	D	Average quantity served			Operational-UB gap		Best solution-UB gap	
		Oper. policy	Best solution	Upper bound	Avg.	Std. dev.	Avg.	Std. dev.
1	3	175.97	176.03	201.33	14.42%	3.02%	14.39%	3.01%
3	3	143.89	143.96	157.53	9.54%	2.87%	9.49%	2.90%
7	2	136.17	136.45	138.73	1.88%	1.27%	1.67%	1.31%
14	2	105.85	106.80	107.85	1.87%	1.75%	0.95%	1.58%

are within 2% of our operational benchmark. For the three-dispatch zones, the average bounds are still fairly close to the operational benchmark, with gaps of under 15% and 10% for zones 1 and 3, respectively. These wider gaps are likely due to the computational difficulty of finding improved solutions for the relatively larger three-dispatch instances, suggesting that the true gaps may be significantly lower. These results are consistent with *a posteriori* bounds calculated with respect to total dispatch time for similar operational policies, and are also in line with gaps observed by more sophisticated operational policies (*e.g.* Klapp et al. 2018a, Stroh et al. 2021).

5.1.3 Sensitivity and managerial insights

The continuous approximation fleet sizing approach facilitates tactical decision-making even without constructing partitions and conducting simulations; we illustrate two examples here. System managers may be interested in the effect of shifting the order cutoff time on fleet maintenance and travel costs. A retailer that anticipates increased adoption of their SDD service will be interested in how many vehicles they need to purchase to keep up with demand. Both questions may be answered by performing sensitivity analyses using Algorithm 1 and equation (7) to calculate the value of V_{CA} as the cutoff time and order rate are varied by small increments. Figures 10a and 10b show how the fleet size guideline varies with cutoff time and order rate, respectively.

Figure 10: Fleet size sensitivity, corner depot



(a) V_{CA} vs. cutoff time (rate of 2 orders/hour/sq. mi.)

(b) V_{CA} vs. order rate (2 PM cutoff)

5.2 Real-World Case Study

Until this point, we have assumed that service regions are geographically homogeneous with respect to vehicle travel. Our first computational study demonstrates that the methodological approach presented in Sections 3 and 4 produces high-quality solutions in such homogeneous regions. However, in some real-world environments, travel speeds may vary within the region based on the presence of various types of roads (*e.g.*, highways vs. residential streets). In this second computational study, we illustrate one potential approach for adapting our methods to such real-world service regions.

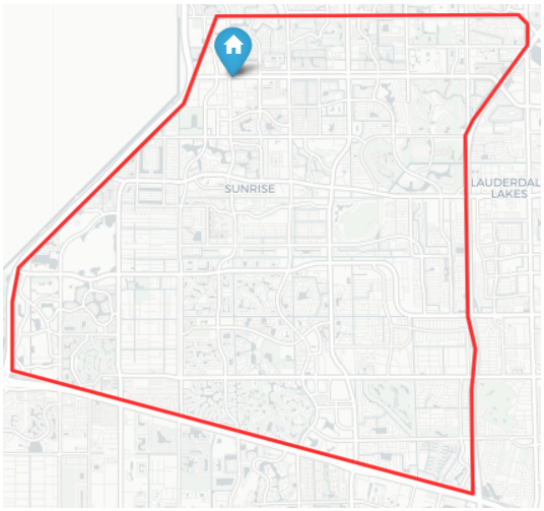


Figure 11: Service region, second computational study

We consider a service region of approximately 46 square miles within Broward County, Florida, depicted in Figure 11 with the depot location marked. The service day again ranges from 8 AM to 8 PM, but we now move the tactical order cutoff time to 4 PM. Time units are taken to be minutes, resulting in $N = 480$ and $T = 720$. As in the previous study, we assume a setup time per dispatch of $\alpha = 5$ minutes and a per-order service time of $\gamma = 2.5$ minutes. Vehicles may dispatch at most $D_{\max} = 3$ times per day and travel along roads at their speed limits. Travel times are asymmetric in general. The order rate per minute per square mile is given by a non-decreasing function $\lambda(t)$ illustrated in Figure 12, which is geographically homogeneous. The average order rate over the interval $[0, N]$ is two orders per hour per square mile as in the previous study.

Our initial goal, as before, is to appropriately characterize the maximum area function for all points in the region. The particular features of this region, however, require us to slightly adapt our approach in two ways. First, we empirically observe that the BHH functional form $\beta\sqrt{An}$ no longer provides a sufficiently accurate approximation for (asymmetric) TSP tour durations. Rather, the growth of the expected TSP duration (not including setup or per-order service times) with n includes both square root and linear components.

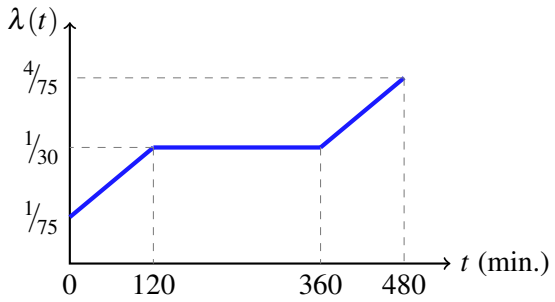


Figure 12: Orders per min. per sq. mi.

This is because, for a fixed zone, a greater number of customers implies that the quickest path between any two consecutive customers on a TSP tour is less likely to involve travel on high-speed major roads – a phenomenon which is not observed when travel times are simply given by a homogeneous metric. To correct for this, we instead use dispatch duration approximations of the form $2\rho + \alpha + \beta\sqrt{An} + \beta^{\text{lin}}n + \gamma n$ for n customers, where $\beta^{\text{lin}} > 0$. Note that simple rearrangement gives $2\rho + \alpha + \beta\sqrt{An} + (\beta^{\text{lin}} + \gamma)n$, which maintains the fundamental structure of the previous dispatch time functions. Thus, our stated theoretical results and algorithms remain valid.

The region is bounded to the west, east, and south by highways with vehicle speed limits up to 70 mph. In contrast, most roads in the region are minor residential streets with speed limits of 30 mph or lower. Major non-highway roads are spread throughout the region with intermediate speed limits. The varying frequency of different types of roads at different parts of the region implies that average vehicle speeds also vary between different parts of the region. As such, assuming identical constants β and β^{lin} throughout the region could lead to a coarse approximation. We instead estimate routing constants at ten separate representative points within the region. To define a dispatch function at any other point r, we linearly interpolate between the routing constants associated with these representative points. Given a dispatch function for any point r, we can then calculate the maximum area $\mathcal{A}(r)$ in the usual fashion.

Specific computational details regarding routing constant estimation and interpolation are provided in Appendix C. Our approach for fitting dispatch functions in this study was developed specifically for the region at hand to demonstrate the practical applicability of our theoretical results, and is not necessarily a prescriptive procedure for use in every region. For example, β^{lin} constants may not always be required to produce accurate estimations of expected tour duration.

5.2.1 Fleet sizing and partitioning

Once we have defined $\mathcal{A}(r)$ across the region, our fleet sizing and partitioning schemes proceed as before. Integrating $1/\mathcal{A}(r)$ over the region implies a minimum fleet size of $V_{\text{CA}} = 7.33$ vehicles. For comparison, the fleet size guidelines are 14.37 and 8.35 for $D_{\max} = 1$ and 2, respectively. We partition the region into $\lceil V_{\text{CA}} \rceil = 8$ zones. Because most roads in the region are oriented north-south or east-west, we continue to use diamond-shaped disks in the automated sliding procedure. This procedure terminates in 995 total iterations in 2 minutes, 7 seconds of runtime with a final shrinkage coefficient of $\kappa = 0.5314$. The disk centers of the resulting output are used as the initial WCVT generator locations.

To construct the partition, the region was discretized into 4,735 blocks via a grid comprised of 0.1 mile $\times$ 0.1 mile squares. We stopped the WCVT balancing process when every zone was connected, all CLFs fell within the range $[0.89, 0.94]$, and the largest distance between any zone's generator and centroid was below 0.1 miles. The process terminated after 57 iterations in 2 minutes, 58 seconds of runtime. Table C3 in Appendix C records the evolution of the WCVT process, and Table C1 in Appendix C lists the parameters used in both phases of the partitioning procedure. The final partition is illustrated in Figure 13 with zone centroids marked.

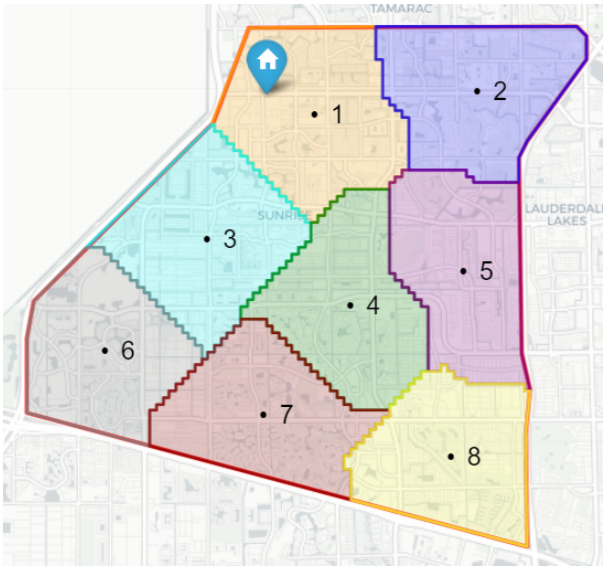


Figure 13: Partitioned service region, second computational study

Zone areas range from 5.37 to 6.33 square miles. As in the prior study, the final average CLF of 0.9135 nearly coincides with the ratio between the computed minimum fleet size and the actual fleet size used, $V_{\text{CA}}/18 \approx 0.9165$. Unlike the prior study, three dispatches are required to serve each zone.

5.2.2 Simulation results

We again simulate 100 service day realizations with the same operational policy, albeit with asymmetric TSPs due to the nature of the underlying road network. Customer locations are generated uniformly at random via VeRoViz and assigned to their closest location on the road network. Distances are calculated with Openrouteservice via VeRoViz. For the simulated cases in which orders are cut off after N, we assume that the order rate for all $t > N$ is $\lambda(t) = \lambda(N) = 4/75$ orders per minute per square mile.

Based on our tactical cutoff time of 4 PM, we expect to serve 733 orders per day in the region. In our simulations, we are able to serve 763 orders per day on average, an improvement of 4.1%. As we would expect based on zone CLFs, every zone cuts off orders later than 4 PM on average; these average simulated cutoff times range from 4:08 PM (zone 7) to 4:23 PM (zone 5). Each zone cuts off orders at or later than 4 PM an average of 72.88 days (of the 100 service days simulated). These results are similar to those of our first computational study, which demonstrates the practical value of our fleet sizing and partitioning approach on real-world road networks and when order rates vary over time. Detailed simulation results for

each zone are given in Appendix C.

6. Conclusions

In this paper, we study the tactical problem of partitioning a single-depot SDD service region into single-vehicle zones. Using continuous approximations on order arrivals and routing times, we first study the sub-problem of independently maximizing the area of a zone served by a single vehicle. We characterize the structure of area-maximizing vehicle dispatch policies, which allows for the calculation of maximum zone areas as a function of the travel time from the zone to the depot. We show how fleet size guidelines can be derived directly from these optimal area functions. We also demonstrate how optimal area functions can be used to create feasible partitions of the service region, extending existing partitioning approaches (Galvão et al. 2006, Ouyang 2007, Ouyang and Daganzo 2006) to the modern context of SDD. We provide two examples of the fleet sizing and partitioning scheme on service regions with different features. In simulations where discrete orders arrive stochastically, we show that a simple operational dispatching policy combined with our partitions produces solutions that meet order cutoff time targets in a vast majority of daily instances. Because of the straightforward method for fleet sizing and the computationally inexpensive partitioning procedure, our technique lends itself to gaining practical tactical insights. Overall, our approach creates an actionable plan for producing well designed vehicle routing zones that satisfy system requirements.

Our work presents multiple avenues for further exploration. An area for future work is the analysis of optimal single-vehicle dispatching policies for extensions of our area-maximization problem. Such extensions may include orders placed before the start of the service day, spatially inhomogeneous order arrivals, capacitated vehicles, and time-dependent vehicle speeds. The last of these potential extensions represents a point of consideration for future applications of continuous approximation methods in other contexts. Our optimal tactical dispatching policies rely on the concavity and monotonicity of BHH-style dispatch functions, and our partitioning approach assumes each zone is served by a single vehicle; however, more complex partitioning schemes involving multiple vehicles working in tandem to serve all orders within a zone may lead to increased efficiency. While Stroh et al. (2021) briefly discuss a heuristic dispatching policy for arbitrarily-sized fleets working within the same region, continued work in this area could focus on deriving optimal area-maximizing and cost-minimizing policies to facilitate such partitions. At the operational level, future work may consider the dynamic assignment of orders across zone boundaries to improve the flexibility of the SDD system or further reduce the number of vehicles required.

Acknowledgements

Dipayan Banerjee's work was supported by the U.S. National Science Foundation's Graduate Research Fellowship (DGE-1650044) and the U.S. Federal Highway Administration's Eisenhower Transportation Research Fellowship. The authors thank Sara Reed for useful discussion regarding the computational studies and the University at Buffalo's Optimator Lab for their development of the open source VeRoViz package. The authors thank the Associate Editor and two anonymous referees for their constructive feedback.

References

Adobe Analytics (2020) Adobe Digital Economy Index. URL https://www.adobe.com/content/dam/www/us/en/experience-cloud/digital-insights/pdfs/adobe_analytics-digital-economy-index-2020.pdf.

Amazon (2018) Amazon expands free shipping to everyone for the holiday season. URL https://press.aboutamazon.com/news-releases/news-release-details/amazon-expands-free-shipping-everyone-holiday-season.

Ansari S, Başdere M, Li X, Ouyang Y, and Smilowitz K (2018) Advancements in continuous approximation models for logistics and transportation systems: 1996 – 2016. *Transportation Research Part B*, 107:229–252.

Applegate D, Bixby R, Chvátal V, and Cook W (2007) *The Traveling Salesman Problem: A Computational Study*. Princeton University Press.

Beardwood J, Halton J, and Hammersley J (1959) The shortest path through many points. *Mathematical Proceedings of the Cambridge Philosophical Society*, 55(4):299–327.

Bergmann F, Wagner S, and Winkenbach M (2020) Integrating first-mile pickup and last-mile delivery on shared vehicle routes for efficient urban e-commerce distribution. *Transportation Research Part B*, 131:26–62.

Bolgert S (2019) tripy 1.0.0. URL https://pypi.org/project/tripy/.

Bramel J and Simchi-Levi D (1995) A location based heuristic for general routing problems. *Operations Research*, 43(4):649–660.

Brent R (1973) *Algorithms for Minimization Without Derivatives*. Prentice-Hall.

Carlsson J (2012) Dividing a territory among several vehicles. *INFORMS Journal on Computing*, 24(4):565–577.

Carlsson J and Delage E (2013) Robust partitioning for stochastic multivehicle routing. *Operations Research*, 61(3):727–744.

Carlsson J and Jia F (2013) Euclidean hub-and-spoke networks. *Operations Research*, 61(6):1360–1382.

Carlsson J and Song S (2018) Coordinated logistics with a truck and a drone. *Management Science*, 64(9):4052–4069.

Cattaruzza D, Absi N, and Feillet D (2016a) The multi-trip vehicle routing problem with time windows and release dates. *Transportation Science*, 50(2):676–693.

Cattaruzza D, Absi N, and Feillet D (2016b) Vehicle routing problems with multiple trips. *4OR*, 14:223–259.

Chen X, Ulmer M, and Thomas B (2021) Deep Q-learning for same-day delivery with vehicles and drones. *European Journal of Operational Research*, in press.

Chen X, Wang T, Ulmer M, and Thomas B (2020) Same-day delivery with fairness. URL `https://arxiv.org/abs/2007.09541`.

Côté JF, de Queiroz TA, Gallesi F, and Iori M (2019) Optimization methods for the same-day delivery problem. In Paolucci M, Sciomachen A, and Uberti P, eds., *Advances in Optimization and Decision Science for Society, Services and Enterprises*, pp. 335–349. Springer.

Daganzo C (1984) The length of tours in zones of different shapes. *Transportation Research Part B*, 188(2):135–145.

Daganzo C (2005) *Logistics Systems Analysis*. Springer, 4th edition.

Dayarian I, Savelsbergh M, and Clarke JP (2020) Same-day delivery with drone resupply. *Transportation Science*, 54(1):229–249.

Du Q, Faber V, and Gunzburger M (1999) Centroidal Voronoi tessellations: applications and algorithms. *SIAM Review*, 41(4):637–676.

Erera A (2000) *Design of Large-Scale Logistics Systems for Uncertain Environments*. Ph.D. thesis, University of California, Berkeley.

Franceschetti A, Honhon D, Laporte G, van Woensel T, and Fransoo J (2017a) Strategic fleet planning for city logistics. *Transportation Research Part B*, 95:19–40.

Franceschetti A, Jabali O, and Laporte G (2017b) Continuous approximation models in freight distribution management. *TOP*, 25(3):413–433.

Galvão L, Novaes A, Souza De Cursi J, and Souza J (2006) A multiplicatively-weighted Voronoi diagram approach to logistics districting. *Computers and Operations Research*, 33(1):93–114.

Heidelberg Institute for Geoinformation Technology (2021) Openrouteservice. URL `https://openrouteservice.org/`.

Johnson D, McGeoch L, and Rothberg E (1996) Asymptotic experimental analysis for the Held-Karp traveling salesman bound. In *Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 341–350.

Klapp M, Erera A, and Toriello A (2018a) The dynamic dispatch waves problem for same-day delivery. *European Journal of Operational Research*, 271(2):519–534.

Klapp M, Erera A, and Toriello A (2018b) The one-dimensional dynamic dispatch waves problem. *Transportation Science*, 52(2):402–415.

Klapp M, Erera A, and Toriello A (2020) Request acceptance in same-day delivery. *Transportation Research Part E*, 143:102083.

Lassiter K (2011) What are census blocks? URL `https://www.census.gov/newsroom/blogs/random-samplings/2011/07/what-are-census-blocks.html`.

Lei H, Laporte G, and Guo B (2012) Districting for routing with stochastic customers. *EURO Journal on Transportation and Logistics*, 1(1-2):67–85.

Merchán D and Winkenbach M (2019) An empirical validation and data-driven extension of continuum approximation approaches for urban route distances. *Networks*, 73(4):418–433.

Newell G and Daganzo C (1986a) Design of multiple-vehicle delivery tours - I a ring-radial network. *Transportation Research Part B*, 20B(5):345–363.

Newell G and Daganzo C (1986b) Design of multiple vehicle delivery tours - II other metrics. *Transportation Research Part B*, 20B(5):365–376.

Okabe A, Boots B, Sugihara K, and Chiu SN (2000) *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, 2nd edition.

Ouyang Y (2007) Design of vehicle routing zones for large-scale distribution systems. *Transportation Research Part B*, 41(10):1079–1093.

Ouyang Y and Daganzo C (2006) Discretization and validation of the continuum approximation scheme for terminal system design. *Transportation Science*, 40(1):89–98.

Oyola J, Arntzen H, and Woodruff D (2017) The stochastic vehicle routing problem, a literature review, part II: solution methods. *EURO Journal on Transportation and Logistics*, 6:349 – 388.

Oyola J, Arntzen H, and Woodruff D (2018) The stochastic vehicle routing problem, a literature review, part I: models. *EURO Journal on Transportation and Logistics*, 7:193 – 221.

Paradiso R, Roberti R, Laganá D, and Dullaert W (2020) An exact solution framework for multitrip vehicle-routing problems with time windows. *Operations Research*, 68(1):180–198.

Peng L and Murray C (2020) VeRoViz: A vehicle routing visualization toolkit. URL `https://ssrn.com/abstract=3746037`.

Pillac V, Gendreau M, Guéret C, and Medaglia A (2013) A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225:1–11.

Psaraftis H, Wen M, and Kontovas C (2016) Dynamic vehicle routing problems: three decades and counting. *Networks*, 67(1):3–31.

Reyes D, Erera A, and Savelsbergh M (2018) Complexity of routing problems with release dates and deadlines. *European Journal of Operational Research*, 266(1):29–34.

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

Schlömer N (2020) quadpy 0.14.11. URL `https://pypi.org/project/quadpy/`.

SciPy (2019) Optimization and root finding (scipy.optimize). URL `https://docs.scipy.org/doc/scipy/reference/optimize.html`.

Stroh A, Erera A, and Toriello A (2021) Tactical design of same-day delivery systems. URL `https://doi.org/10.1287/mnsc.2021.4041`. *Management Science*, forthcoming.

Thomas L (2019) Target expands same-day shipping option in the latest move in the delivery wars with Walmart and Amazon. URL `https://www.cnbc.com/2019/06/12/target-expands-same-day-delivery-option-in-battle-with-walmart-amazon.html`.

Ulmer M (2017) Delivery deadlines in same-day delivery. *Logistics Research*, 10(1):1–15.

Ulmer M, Goodson J, Mattfeld D, and Hennig M (2019a) Offline–online approximate dynamic programming for dynamic vehicle routing with stochastic requests. *Transportation Science*, 53(1):185–202.

Ulmer M, Mattfeld D, and Köster F (2018) Budgeting time for dynamic vehicle routing with stochastic customer requests. *Transportation Science*, 52(1):20–37.

Ulmer M and Streng S (2019) Same-day delivery with pickup stations and autonomous vehicles. *Computers and Operations Research*, 108:1–19.

Ulmer M and Thomas B (2018) Same-day delivery with heterogeneous fleets of drones and vehicles. *Networks*, 72(4):475–505.

Ulmer M, Thomas B, and Mattfeld D (2019b) Preemptive depot returns for dynamic same-day delivery. *EURO Journal on Transportation and Logistics*, 8(4):327–361.

United States Census Bureau (2020) Quarterly retail e-commerce sales 1st quarter 2020. URL `https://www2.census.gov/retail/releases/historical/ecomm/20q1.pdf`.

van Heeswijk W, Mes M, and Schutten M (2019) The delivery dispatching problem with time windows for urban consolidation centers. *Transportation Science*, 53(1):203–221.

Voccia S, Campbell A, and Thomas B (2019) The same-day delivery problem for online purchases. *Transportation Science*, 53(1):167–184.

Xiao H and Gimbutas Z (2010) A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions. *Computers & Mathematics with Applications*, 59(2):663–676.

Yao B, McLean C, and Yang H (2019) Robust optimization of dynamic route planning in same-day delivery networks with one-time observation of new demand. *Networks*, 73(4):434–452.

Zhong H, Hall R, and Dessouky M (2007) Territory planning and vehicle dispatching with driver learning. *Transportation Science*, 41(1):74–89.

Appendix A Omitted Proofs

A.1 Proof of Lemma 1

Consider an area $A > 0$ and an associated feasible dispatching policy $P = \big((t_1, \tau_1), \ldots, (t_D, \tau_D)\big)$ that involves waiting. To avoid the trivial case, assume $D > 1$. There exists some $d \in [D]$ such that there is waiting after dispatch d.

If $d = 1$, set $P' = \big((t'_1, \tau'_1), \ldots, (t'_D, \tau'_D)\big) = \big((t_1, \tau_1), \ldots, (t_D, \tau_D)\big)$. Suppose instead that $d \geq 2$. Let $\varepsilon > 0$ equal the waiting time after dispatch d. Define $\bar{t}_d = t_d + \frac{\varepsilon}{2}$, then define $\bar{t}_{d-1} = t_{d-1} + \frac{\varepsilon}{4}$, and so on until $\bar{t}_1 = t_1 + \frac{\varepsilon}{2^d}$. Set $P' = \big((t'_1, \tau'_1), \ldots, (t'_D, \tau'_D)\big) = \big((\bar{t}_1, \tau_1), \ldots, (\bar{t}_d, \tau_d), (t_{d+1}, \tau_{d+1}), \ldots, (t_D, \tau_D)\big)$. Observe that P' is identical to P except that the first d departure times have been shifted such that there is waiting after each of the first d dispatches. If $d = D$, set $\hat{P} = P'$; this is the desired policy. Suppose instead that $d < D$. Consider two cases.

Case I: $\tau'_{d+1} > 0$. Because there is waiting after the d-th dispatch of P', by the continuity of f_ρ there exists some $\delta \in (0, \tau'_{d+1})$ such that $(t'_d + \delta) + f_\rho(A, \tau'_d + \delta) < t'_{d+1}$. Thus, if we replace t'_d by $t'_d + \delta$, τ'_d by $\tau'_d + \delta$, and τ'_{d+1} by $\tau'_{d+1} - \delta$, then there is waiting after both the d-th dispatch and the $(d+1)$-th dispatch.

Case II: $\tau'_{d+1} = 0$. Let the waiting time after the d-th dispatch of P' be $\mu > 0$. If we replace t'_{d+1} by $t'_{d+1} - \frac{\mu}{2}$, then we have waiting after both the d-th dispatch and the $(d+1)$-th dispatch.

We can repeat this pairwise replacement procedure via the relevant case for the $(d+1)$-th dispatch and the $(d+2)$-th dispatch, followed by the $(d+2)$-th dispatch and the $(d+3)$-th dispatch, and so on until the $(D-1)$-th dispatch and the D-th dispatch; denote the resulting policy $\hat{P}$. By our construction, $\hat{P}$ is a D-dispatch policy feasible for A with waiting after every dispatch. □

A.2 Proof of Theorem 3

For notational convenience, let $\hat{D} = D_{\max}(\rho)$. Let $\hat{D} \geq 2$ and $2 \leq D \leq \hat{D}$ such that $\mathcal{A}^+_{\hat{D}}(\rho) = \mathcal{A}_D(\rho) = A > 0$. Let $\mathcal{P}$ represent the set of all D-dispatch policies feasible for A such that for some $d \in [D]$, the corresponding constraint (1b) holds at strict inequality. For the purposes of contradiction, suppose that $\mathcal{P}$ is nonempty. For all policies $P \in \mathcal{P}$, define k_P as the smallest dispatch index such that the corresponding constraint (1b) holds at strict inequality. Define

$$P_0 = \operatorname*{argmin}_{P \in \mathcal{P}} \{k_P\}.$$

Let $P_0 = \big((t_1,\tau_1),\ldots,(t_D,\tau_D)\big)$ and define $k = k_{P_0}$. Because A is the optimal area, Theorem 2 implies $t_1 = \tau_1$, so $k \geq 2$. By definition, $\sum_{i=1}^{d} \tau_i = t_d$ for all $d < k$, and $\sum_{i=1}^{k} \tau_i < t_k$. Define $V = \sum_{i=1}^{k} \tau_i$, the total accumulation time of the first k dispatches, and $W = t_k + f_\rho(A,\tau_k)$, the return time of the k-th dispatch. If $k = D$, then $W = T$; otherwise, W equivalently represents the departure time of the $(k+1)$-th dispatch. Note that by Theorem 2, for all $d \in [k-1]$, the return time of dispatch d is $t_d + f_\rho(A,\tau_d) = t_{d+1}$. This implies $f_\rho(A,\tau_d) = \tau_{d+1}$ for all $d \in [k-2]$. Assume that $\tau_1,\ldots,\tau_{k-1} > 0$ and consider three cases.

Case I: $\tau_{k-1} \leq \tau_k$. Because f_ρ is continuous, strictly concave, and increasing in τ for a fixed area, there exists some $0 < \varepsilon < \tau_{k-1}$ such that $t_{k-1} + f_\rho(A,\tau_{k-1}-\varepsilon) \geq V$ and $t_{k-1} + f_\rho(A,\tau_{k-1}-\varepsilon) + f_\rho(A,\tau_k+\varepsilon) < W$. This follows from the fact that $\tau_{k-1} \leq \tau_k$ implies $f_\rho(A,\tau_{k-1}-\varepsilon) + f_\rho(A,\tau_k+\varepsilon) < f_\rho(A,\tau_{k-1}) + f_\rho(A,\tau_k)$. Define the policy

$$P_1 = \big((t_1,\tau_1),\ldots,(t_{k-1},\tau_{k-1}-\varepsilon),(t_{k-1}+f_\rho(A,\tau_{k-1}-\varepsilon),\tau_k+\varepsilon),(t_{k+1},\tau_{k+1}),\ldots,(t_D,\tau_D)\big).$$

Note that P_1 involves waiting after the k-th dispatch (as illustrated in Figure A2) and is feasible for A, so A is not the optimal D-dispatch area, a contradiction.

Figure A1: $(k-1)$-th and k-th dispatches in policy P_0, Case I

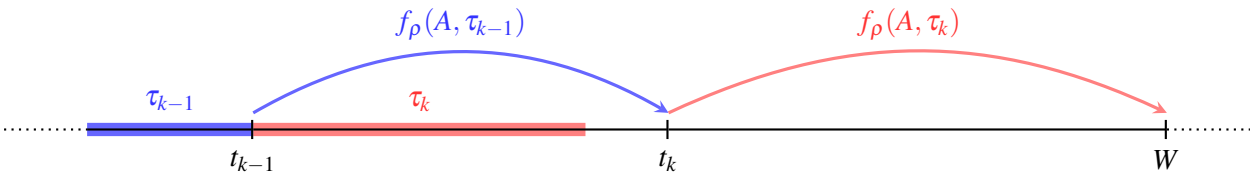
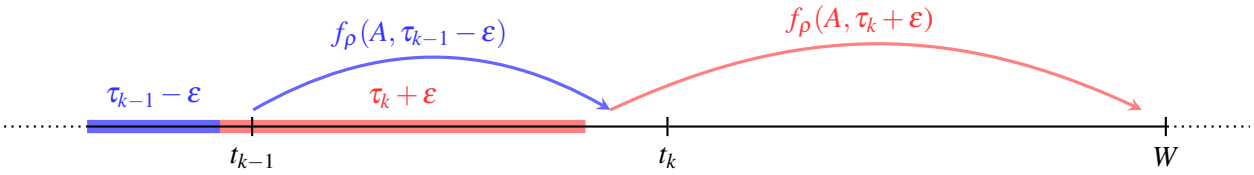


Figure A2: $(k-1)$-th and k-th dispatches in policy P_1



Case II: $\tau_{k-1} > \tau_k$ and $f_\rho(A,\tau_k) \geq \tau_k$. Because $t_{k-1} + \tau_k = V$, it follows that $t_{k-1} + f_\rho(A,\tau_k) \geq V$. As $t_{k-1} + f_\rho(A,\tau_{k-1}) + f_\rho(A,\tau_k) = W$, trivially $t_{k-1} + f_\rho(A,\tau_k) + f_\rho(A,\tau_{k-1}) = W$. Therefore, the policy

$$P_2 = \big((t_1, \tau_1), \ldots, (t_{k-1}, \tau_k), (t_{k-1} + f_\rho(A, \tau_k), \tau_{k-1}), (t_{k+1}, \tau_{k+1}), \ldots, (t_D, \tau_D)\big)$$

is feasible for A. Observe that P_2 was constructed by "swapping" the dispatch accumulation times τ_{k-1} and τ_k (as illustrated in Figure A4). Because $\sum_{i=1}^{k-1} \tau_i = t_{k-1}$ and $\tau_{k-1} > \tau_k$, we have that $\tau_k + \sum_{i=1}^{k-2} \tau_i < t_{k-1}$. This implies that $k_{P_2} < k$, which contradicts the definitions of k and P_0.

Figure A3: $(k-1)$-th and k-th dispatches in policy P_0, Case II

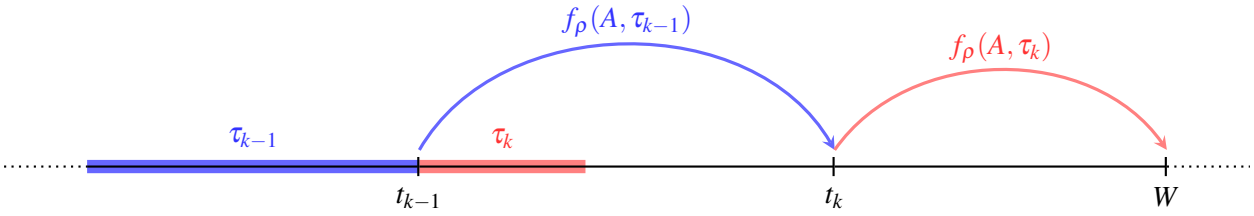
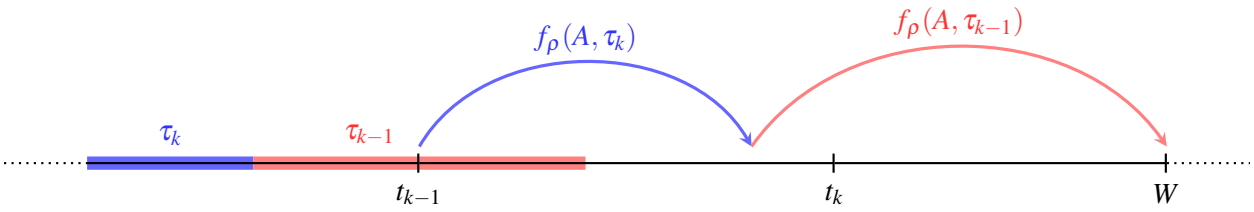


Figure A4: $(k-1)$-th and k-th dispatches in policy P_2



Case III: $\tau_{k-1} > \tau_k$ and $f_\rho(A, \tau_k) < \tau_k$. Define $f_\rho^{(0)}(A, \tau) = \tau$ and $f_\rho^{(j)}(A, \tau) = f_\rho(A, f_\rho^{(j-1)}(A, \tau))$ for all $j \geq 1$. For all $d \in [k-1]$, $\tau_d = f_\rho^{(d-1)}(A, \tau_1)$. Additionally, define the function h by

$$h(\tau) = \sum_{j=0}^{k-2} f_\rho^{(j)}(A, \tau).$$

In other words, $h(\tau)$ represents the total accumulation time of the first $k-1$ dispatches if τ is the accumulation time of the first dispatch. Note that $t_{k-1} = h(\tau_1)$ and that h is an increasing continuous function. Let $\mu = t_{k-1} - h(0)$, and let ν be such that $f_\rho(A, \tau_k + \nu) = W - V$. Since $\tau_1 > 0$, we have that $\mu > 0$, and since $f_\rho(A, \tau_k) < W - V$, we have that $\nu > 0$. Define $\delta = \min\{\mu, \nu\}$. Because f_ρ is strictly increasing and concave in τ for a fixed A and $f_\rho(A, \tau_k) < \tau_k$, it follows that $f_\rho(A, \tau_k + \delta) < f_\rho(A, \tau_k) + \delta$.

Because $\tau_{k-1} > \tau_k$, it also follows inductively that $\tau_1 > \tau_2 > \cdots > \tau_{k-1}$. Then, there exist $\varepsilon_1, \ldots, \varepsilon_{k-1} > 0$

such that

(i) $h(\tau_1 - \varepsilon_1) = t_{k-1} - \delta$, and

(ii) $f_\rho(A, \tau_j - \varepsilon_j) = \tau_{j+1} - \varepsilon_{j+1}$ for all $j \in [k-2]$.

Additionally, $f_\rho(A, \tau_{k-1} - \varepsilon_{k-1}) < f_\rho(A, \tau_{k-1})$. Therefore,

$$f(A, \tau_{k-1} - \varepsilon_{k-1}) + f(A, \tau_k + \delta) < \delta + f(A, \tau_{k-1}) + f(A, \tau_k). \tag{A1}$$

In other words, we appropriately modify the first $k-1$ accumulation times such that the total accumulation times of the first $k-1$ dispatches is δ lower than in P_0, and the accumulation time of the k-th dispatch is δ greater than in P_0. By (A1), it is possible to complete both the new $(k-1)$-th and k-th dispatches in the time interval $[t_{k-1} - \delta, W]$, as $W - t_{k-1} = f_\rho(A, \tau_k + \delta) < \delta + f_\rho(A, \tau_{k-1}) + f_\rho(A, \tau_k)$.

Figure A5: First $k = 5$ dispatches in policy P_0, Case III

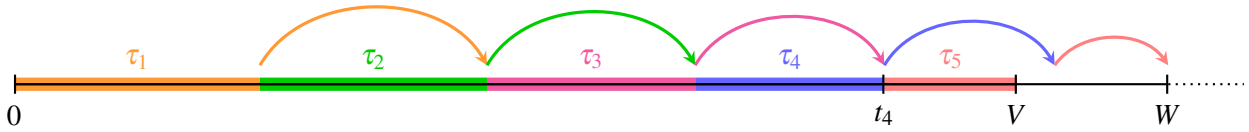
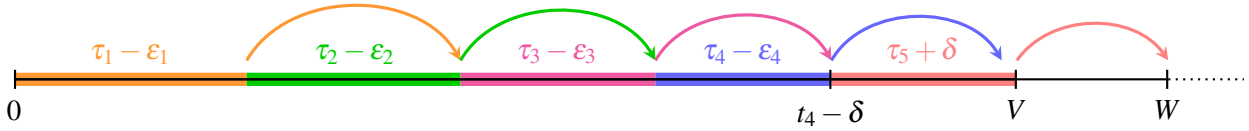


Figure A6: First $k = 5$ dispatches in policy P_3 with waiting after the 4th dispatch



Thus, the policy

$$\begin{aligned} P_3 = \big(&(t_1 - \varepsilon_1, \tau_1 - \varepsilon_1), (t_2 - \varepsilon_1 - \varepsilon_2, \tau_2 - \varepsilon_2), \ldots, \\ &(t_{k-1} - \delta, \tau_{k-1} - \varepsilon_{k-1}), (W - f_\rho(A, \tau_k + \delta), \tau_k + \delta), (t_{k+1}, \tau_{k+1}), \ldots, (t_D, \tau_D) \big) \end{aligned}$$

is feasible for A. Figures A5 and A6 illustrate policies P_0 and P_3 for an example with $k = 5$. Because (A1) is a strict inequality, P_3 includes waiting after the $(k-1)$-th dispatch. Therefore, A is not optimal for D dispatches, a contradiction.

Hence, when $\tau_1, \ldots, \tau_{k-1} > 0$, we have arrived at a contradiction in every case. Suppose instead that one of the first $k-1$ accumulation times for P_0 is zero. If there are $m > 0$ dispatches with zero quantity among the first $k-1$ dispatches (assuming $m \leq D-2$ to avoid triviality), we can remove all m of these dispatches to be left with one of the following:

- some $(D-m)$-dispatch policy feasible for A that involves waiting; this implies $A < \mathcal{A}_{D-m}(\rho) \leq \mathcal{A}^+_{\hat{D}}(\rho)$, a contradiction, or
- some $(D-m)$-dispatch policy feasible for A such that $\sum_{i=1}^{d} \tau_i < t_i$ for some d. Because $\mathcal{A}_D(\rho) = \mathcal{A}^+_{\hat{D}}(\rho)$, we also have $\mathcal{A}_{D-m}(\rho) = \mathcal{A}^+_{\hat{D}}(\rho)$. However, in the previous part of the proof, we showed that $\mathcal{A}_{D-m}(\rho) < \mathcal{A}^+_{\hat{D}}(\rho)$, a contradiction.

Thus, we have shown that $\mathcal{P} = \varnothing$ as desired. □

A.3 Derivation of equations (3a) and (3b)

We have that $g_\rho(A,t) = \tau$ such that $f_\rho(A,\tau) = t$. When $\gamma = 0$, we wish to solve $t = 2\rho + \beta A\sqrt{\tau}$ for τ. Simply rearranging this equation gives

$$\tau = \left(\frac{t-2\rho}{\beta A}\right)^2, \tag{A2}$$

as desired. When $\gamma > 0$, we need to solve $t = 2\rho + \gamma A\tau + \beta A\sqrt{\tau}$ for τ. Making the substitution $x = \sqrt{\tau}$ and rearranging gives

$$\gamma A x^2 + \beta A x + (2\rho - t) = 0. \tag{A3}$$

Solving this quadratic equation for x gives

$$x = \frac{-\beta A \pm \sqrt{(\beta A)^2 - 4\gamma A(2\rho - t)}}{2\gamma A}, \tag{A4}$$

which implies

$$\tau = \left(\frac{-\beta A \pm \sqrt{(\beta A)^2 + 4\gamma A(t - 2\rho)}}{2\gamma A}\right)^2. \tag{A5}$$

The boundary condition $g_\rho(A, 2\rho) = 0$ implies (3b) as desired. □

A.4 Proof of Proposition 4

Let $\mathcal{A}^+_D(\rho) = \mathcal{A}_D(\rho) > 0$. To first prove the forward implication, suppose that $\mathcal{A}_{D+1}(\rho) \geq \mathcal{A}^+_D(\rho)$. Then, by definition, $\mathcal{A}_{D+1}(\rho) = \mathcal{A}^+_{D+1}(\rho)$; consider the optimal $(D+1)$-dispatch policy, with first departure time

t_1, associated with this area. By Theorems 2 and 3, the $(D+1)$-th dispatch of this policy departs the depot at time N, so

$$\sum_{d=0}^{D} f_\rho^{(d)}(\mathcal{A}_{D+1}(\rho), t_1) = N. \tag{A6}$$

Since $t_1 \geq 0$,

$$\sum_{d=0}^{D} f_\rho^{(d)}(\mathcal{A}_D^+(\rho), 0) = \sum_{d=1}^{D} f_\rho^{(d)}(\mathcal{A}_D^+(\rho), 0) \leq N \tag{A7}$$

because f_ρ and its compositions are component-wise non-decreasing.

To prove the reverse implication, suppose that condition (5) holds. Consider the optimal D-dispatch policy with first departure time t_1'. By Theorems 2 and 3, the D-th dispatch of this policy departs the depot at time N, so

$$\sum_{d=0}^{D-1} f_\rho^{(d)}(\mathcal{A}_D^+(\rho), t_1') = N. \tag{A8}$$

Observe that $t_1' \geq 2\rho$ must hold, otherwise

$$\sum_{d=0}^{D-1} f_\rho^{(d)}(\mathcal{A}_D^+(\rho), 2\rho) > N, \tag{A9}$$

which is a contradiction since

$$\sum_{d=0}^{D-1} f_\rho^{(d)}(\mathcal{A}_D^+(\rho), 2\rho) = \sum_{d=0}^{D} f_\rho^{(d)}(\mathcal{A}_D^+(\rho), 0) = \sum_{d=1}^{D} f_\rho^{(d)}(\mathcal{A}_D^+(\rho), 0) \leq N. \tag{A10}$$

Therefore, we can feasibly insert an empty (zero-quantity) dispatch at time 0 in the optimal D-dispatch policy, so $\mathcal{A}_{D+1}(\rho) \geq \mathcal{A}_D^+(\rho)$.

A.5 Proof of Proposition 5

Let $A = \mathcal{A}_D(\rho)$ for notational convenience. Observe that, when the area is fixed at A, f_ρ is concave and increasing in τ. If $\gamma A \geq 1$, then for all $\tau > 0$, $f_\rho(A, \tau) > \tau$.

If not, then there exists exactly one value τ^* such that $f_\rho(A, \tau^*) = \tau^*$. Additionally,

- if $\tau < \tau^*$, then $\tau < f_\rho(A, \tau) < \tau^*$, and
- if $\tau > \tau^*$, then $\tau > f_\rho(A, \tau) > \tau^*$.

In all cases, the relevant properties imply that any policy with the "optimal structure" must have monotonic (increasing, constant, or decreasing) dispatch accumulation times.

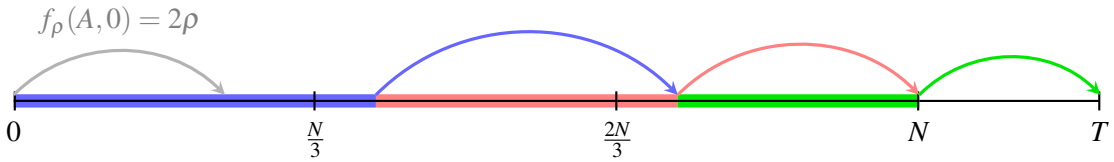
Suppose $\frac{N}{T} > \frac{D}{D+1}$. Then, the mean accumulation time across all dispatches $\frac{1}{D}\sum_{d=1}^{D}\tau_d$ is equal to $\frac{1}{D+1}$. Because $f_\rho(A,\tau_D) = T - N < \frac{1}{D+1}$ and dispatch accumulation times are monotonic, the dispatch accumulation times must be monotonically decreasing. Hence, the dispatch quantities are monotonically decreasing, as desired. Analogous arguments imply the desired results in the cases of $\frac{N}{T} = \frac{D}{D+1}$ and $\frac{N}{T} < \frac{D}{D+1}$. □

A.6 Proof of Proposition 6

Proof for $D = 2$. Consider an area maximization problem with $\alpha = 0$ and $\frac{N}{T} \geq \frac{1}{2}$. Let $\rho \in [0, \frac{T-N}{2})$. It follows that $2\rho < N$. The maximum one-dispatch area $\mathcal{A}_1(\rho)$ is associated with a feasible dispatching policy $\big((t_1,\tau_1)\big)$ such that $t_1 = N$. If a vehicle leaves the depot at $t = 0$ with no orders, it arrives back at the depot before t_1. Thus, $\big((0,0),(t_1,\tau_1)\big)$ is a feasible two-dispatch policy for the area $\mathcal{A}_1(\rho)$ that includes waiting after the first dispatch. Therefore, $\mathcal{A}_2(\rho) > \mathcal{A}_1(\rho)$. The converse is trivial, since if $\frac{N}{T} < \frac{1}{2}$, there exists some $\rho^* \in [0, \frac{T-N}{2})$ such that $4\rho^* > T$, implying $\mathcal{A}_2(\rho^*) = 0 < \mathcal{A}_1(\rho^*)$. □

Proof for $D \geq 3$. We prove the general claim by induction with $D = 2$ as the base case, proved above. Consider an area maximization problem with $\alpha = 0$, $\rho \in [0, \frac{T-N}{2})$ and some $D \geq 3$ such that $\frac{N}{T} \geq \frac{D-1}{D}$. It follows that $2\rho < \frac{N}{D-1}$. Assume, for the purposes of induction, that $\mathcal{A}_{D-1}(\rho) > \mathcal{A}_{D-2} > \cdots > \mathcal{A}_1(\rho)$ since $\frac{D-1}{D-2} < \frac{D}{D-1} \leq \frac{N}{T}$. The maximum $(D-1)$-dispatch area $\mathcal{A}_{D-1}(\rho)$ is associated with some optimal $(D-1)$-dispatch policy $\big((t_1,\tau_1),\ldots,(t_{D-1},\tau_{D-1})\big)$. Because $\mathcal{A}_{D-1}(\rho) = \mathcal{A}^+_{D-1}(\rho)$, by Lemma 5 we have that $\tau_1 \geq \tau_2 \geq \cdots \geq \tau_{D-1}$. Suppose $t_1 = \tau_1 < \frac{N}{D-1}$. Then, $\sum_{d=1}^{D-1}\tau_d < N$, a contradiction, so $t_1 \geq \frac{N}{D-1}$. As such, if a vehicle leaves the depot at $t = 0$ with no orders, it will arrive back at the depot before t_1. The example in Figure A7 illustrates this argument for $D - 1 = 3$: because accumulation times are non-decreasing, $\tau_1 = t_1$ must represent at least $1/3$ of the total accumulation time N. For any $\rho \in [0, \frac{T-N}{2})$, a zero-quantity dispatch at time 0 must return to the depot before $N/3$.

Figure A7: Policy example with $D - 1 = 3$ dispatches



It follows that $\big((0,0),(t_1,\tau_1),\ldots,(t_{D-1},\tau_{D-1})\big)$ is a feasible D-dispatch policy for the area $\mathcal{A}_{D-1}(\rho)$ that includes waiting after the first dispatch. Therefore, $\mathcal{A}_D(\rho) > \mathcal{A}_{D-1}(\rho) > \mathcal{A}_{D-2} > \cdots > \mathcal{A}_1(\rho)$. The converse is again trivial, since if $\frac{N}{T} < \frac{D-1}{D}$, there exists some $\rho^* \in [0, \frac{T-N}{2})$ such that $2D\rho^* > T$, implying $\mathcal{A}_D(\rho^*) = 0 < \mathcal{A}_1(\rho^*)$. □

Appendix B Fleet Sizing for Time-Varying Order Rates

In this section, we relax the assumption that the order rate is constant over time and revisit the zone area maximization and fleet sizing problems. Specifically, we assume that the order rate is non-decreasing over time. While the main theoretical ideas are analogous to those in the constant-rate case, we re-define some notation in order to formally motivate and derive theoretical results in this setting.

Suppose that, for each point in the region R, the order rate per unit area at time t is given by a (Riemann) integrable function $\lambda : [0,N] \to \mathbb{R}_{>0}$. We assume that λ is non-decreasing: for all $0 \leq s < t \leq N$, $\lambda(s) \leq \lambda(t)$. For all $0 \leq s \leq t \leq N$, define

$$\Lambda(s,t) = \int_s^t \lambda(u)du$$

for notational convenience. Because of time-varying order rates, accumulation times τ_d are no longer necessarily proportional to dispatch quantities, and we must define dispatch policies in a more detailed manner. For each dispatch d, let ω_d represent the dispatch quantity prior to scaling by area, i.e., $q_d = A\omega_d$. A D-dispatch policy is characterized by a D-tuple of ordered triplets $\big((t_1,\tau_1,\omega_1),(t_2,\tau_2,\omega_2),\ldots,(t_D,\tau_D,\omega_D)\big)$. Note that, by definition, $\omega_d = \Lambda\left(\sum_{i=1}^{d-1}\tau_i, \sum_{i=1}^{d}\tau_i\right)$ for all $d \in [D]$. However, we include the ω_d variables in dispatch policy definitions for notational clarity.

As before, suppose that the centroid of the zone whose area we wish to maximize is located ρ time units from the depot. Because the quantity of orders on a dispatch d is $A\omega_d$, we can re-define the dispatch time function $f_\rho(A,\omega) = 2\rho + \alpha + \beta A\sqrt{\omega} + \gamma A\omega$. For simplicity, we again assume $\alpha = 0$. This dispatch time function is continuous and increasing in A for a fixed $\omega > 0$. For a fixed $A > 0$, f_ρ is also increasing and strictly concave in ω.

For a fixed $\rho \geq 0$ and $0 < D \leq D_{\max}(\rho)$, assume that the feasibility conditions $2\rho < T - N$ and $2D\rho < T$ are met. We define $\mathcal{A}_D(\rho)$ as the optimal value of the following optimization problem:

$$\mathcal{A}_D(\rho) = \max_{t,\tau,\omega,A} \quad A \tag{B1a}$$

$$\text{s.t. } t_d \geq \sum_{i=1}^{d}\tau_i \qquad \forall d \in [D], \tag{B1b}$$

$$\omega_d = \Lambda\left(\sum_{i=1}^{d-1}\tau_i, \sum_{i=1}^{d}\tau_i\right) \qquad \forall d \in [D], \tag{B1c}$$

$$t_{d+1} \geq t_d + f_\rho(A,\omega_d) \qquad \forall d \in [D-1], \tag{B1d}$$

$$T \geq t_D + f_\rho(A, \omega_D), \tag{B1e}$$

$$\sum_{d=1}^{D} \tau_d = N, \tag{B1f}$$

$$\tau_d, \omega_d \geq 0 \qquad \forall d \in [D]. \tag{B1g}$$

B.1 Optimal policy structure

Due to the lack of assumptions on the behavior of $\lambda(t)$ beyond it being non-decreasing and integrable, the optimization problem above can be prohibitively difficult to solve by standard methods. As in our analysis of the constant-rate setting, we instead seek to characterize properties of area-maximizing policies to derive an efficient solution method. First, we state and prove a result analogous to Lemma 1. Note that the proof of Lemma 8 is virtually identical to that of Lemma 1 aside from some changes in notation for consistency.

Lemma 8. *Consider an area $A > 0$ and an associated feasible policy $P = \big((t_1, \tau_1, \omega_1), \ldots, (t_D, \tau_D, \omega_D)\big)$. If P involves waiting, there exists a feasible policy $\hat{P} = \big((\hat{t}_1, \hat{\tau}_1, \hat{\omega}_1), \ldots, (\hat{t}_D, \hat{\tau}_D, \hat{\omega}_D)\big)$ serving A that includes waiting after every dispatch $d \in [D]$.*

Proof. Consider an area $A > 0$ and an associated feasible dispatching policy $P = \big((t_1, \tau_1, \omega_1), \ldots, (t_D, \tau_D, \omega_D)\big)$ that involves waiting. To avoid the trivial case, assume $D > 1$. There exists some $d \in [D]$ such that there is waiting after dispatch d.

If $d = 1$, set $P' = \big((t'_1, \tau'_1), \ldots, (t'_D, \tau'_D)\big) = \big((t_1, \tau_1), \ldots, (t_D, \tau_D)\big)$. Suppose instead that $d \geq 2$. Let $\varepsilon > 0$ equal the waiting time after dispatch d. Define $\bar{t}_d = t_d + \frac{\varepsilon}{2}$, then define $\bar{t}_{d-1} = t_{d-1} + \frac{\varepsilon}{4}$, and so on until $\bar{t}_1 = t_1 + \frac{\varepsilon}{2^d}$. Set

$$P' = \big((t'_1, \tau'_1, \omega'_1), \ldots, (t'_D, \tau'_D, \omega'_D)\big) = \big((\bar{t}_1, \tau_1, \omega_1), \ldots, (\bar{t}_d, \tau_d, \omega_d), (t_{d+1}, \tau_{d+1}, \omega_{d+1}), \ldots, (t_D, \tau_D, \omega_D)\big).$$

Observe that P' is identical to P except that the first d departure times have been shifted such that there is waiting after each of the first d dispatches. If $d = D$, set $\hat{P} = P'$; this is the desired policy. Suppose instead that $d < D$. Consider two cases.

Case I: $\tau'_{d+1} > 0$ (equivalently, $\omega'_{d+1} > 0$). Because there is waiting after the d-th dispatch of P', by the

continuity of f_ρ and Λ there exists some $\delta \in (0, \tau'_{d+1})$ such that $(t'_d + \delta) + f_\rho\left(A, \omega'_d + \zeta\right) < t'_{d+1}$, where

$$\zeta = \Lambda\left(\sum_{i=1}^{d} \tau'_i, \delta + \sum_{i=1}^{d} \tau'_i\right).$$

Thus, if we replace t'_d by $t'_d + \delta$, τ'_d by $\tau'_d + \delta$, ω'_d by $\omega'_d + \zeta$, τ'_{d+1} by $\tau'_{d+1} - \delta$, and ω'_{d+1} by $\omega'_{d+1} - \zeta$, then there is waiting after both the d-th dispatch and the $(d+1)$-th dispatch.

Case II: $\tau'_{d+1} = 0$. Let the waiting time after the d-th dispatch of P' be $\mu > 0$. If we replace t'_{d+1} by $t'_{d+1} - \frac{\mu}{2}$, then we have waiting after both the d-th dispatch and the $(d+1)$-th dispatch.

We can repeat this pairwise replacement procedure via the relevant case for the $(d+1)$-th dispatch and the $(d+2)$-th dispatch, followed by the $(d+2)$-th dispatch and the $(d+3)$-th dispatch, and so on until the $(D-1)$-th dispatch and the D-th dispatch; denote the resulting policy $\hat{P}$. By our construction, $\hat{P}$ is a D-dispatch policy feasible for A with waiting after every dispatch. □

Similar to the constant-rate case, Lemma 8 implies that optimal dispatching policies do not involve waiting. Furthermore, the area-maximizing policy structure derived for constant order rates remains optimal in this case. Theorem 7, restated from Section 3.4, formalizes this result. The proof is very similar to that of Theorem 3 but requires additional technical analysis at certain stages. We include it separately here and note that the constituent numbered cases in this proof (and their associated figures) correspond directly to the cases in the proof of Theorem 3.

Theorem 7. *Assume $\lambda(t)$ is non-decreasing, strictly positive, and integrable. Let $D \leq D_{\max}(\rho)$. If $\mathcal{A}_D(\rho) = \mathcal{A}^+_{D_{\max}}(\rho)$, the optimal D-dispatch policy satisfies the following conditions:*

(a) the policy does not involve waiting,

(b) each dispatch takes all accumulated, unserved orders at the time of departure, and

(c) the final dispatch departs at N.

Conditions (b) and (c) are equivalent to the following: when $A = \mathcal{A}_D(\rho) = \mathcal{A}^+_{D_{\max}}(\rho)$, constraints (B1b) *hold at equality.*

Proof. Lemma 8 and continuity imply condition (a) as in the proof of Theorem 2. We prove conditions (b) and (c) here. For notational convenience, let $\hat{D} = D_{\max}(\rho)$. Let $\hat{D} \geq 2$ and $2 \leq D \leq \hat{D}$ be such that

$\mathcal{A}_D^+(\rho) = \mathcal{A}_D(\rho) = A > 0$. Let $\mathcal{P}$ represent the set of all D-dispatch policies feasible for A such that for some $d \in [D]$, the corresponding constraint (B1b) holds at strict inequality. For the purposes of contradiction, suppose that $\mathcal{P}$ is nonempty. For all policies $P \in \mathcal{P}$, define k_P as the smallest dispatch index such that the corresponding constraint (B1b) holds at strict inequality. Define

$$P_0 = \operatorname*{argmin}_{P \in \mathcal{P}} \{k_P\}.$$

Let $P_0 = \big((t_1, \tau_1, \omega_1), \ldots, (t_D, \tau_D, \omega_D)\big)$ and define $k = k_{P_0}$. Without loss of generality, assume all parameters are scaled so that $\omega_k = \tau_k$.

Because A is the optimal area, the lack of waiting in optimal policies implies $t_1 = \tau_1$, so $k \geq 2$. By definition, $\sum_{i=1}^{d} \tau_i = t_d$ for all $d < k$, and $\sum_{i=1}^{k} \tau_i < t_k$. Define $V = \sum_{i=1}^{k} \tau_i$, the total accumulation time of the first k dispatches, and $W = t_k + f_\rho(A, \tau_k)$, the return time of the k-th dispatch. If $k = D$, then $W = T$; otherwise, W equivalently represents the departure time of the $(k+1)$-th dispatch. Note that, for all $d \in [k-1]$, the return time of dispatch d is $t_d + f_\rho(A, \omega_d) = t_{d+1}$. This implies $f_\rho(A, \omega_d) = \tau_{d+1}$ for all $d \in [k-2]$. Assume that $\tau_1, \ldots, \tau_{k-1} > 0$, which is equivalent to assuming $\omega_1, \ldots, \omega_{k-1} > 0$, and consider three cases.

Case I: $\omega_{k-1} \leq \omega_k$. Because f_ρ is continuous, strictly concave, and increasing in ω for a fixed area, there exists some $0 < \varepsilon < \omega_{k-1}$ such that $t_{k-1} + f_\rho(A, \omega_{k-1} - \varepsilon) \geq V$ and $t_{k-1} + f_\rho(A, \omega_{k-1} - \varepsilon) + f_\rho(A, \omega_k + \varepsilon) < W$. This follows from the fact that $\omega_{k-1} \leq \omega_k$ implies $f_\rho(A, \omega_{k-1} - \varepsilon) + f_\rho(A, \omega_k + \varepsilon) < f_\rho(A, \omega_{k-1}) + f_\rho(A, \omega_k)$ due to strict concavity.

Choose $0 < \delta < \tau_{k-1}$ such that

$$\Lambda\left(\sum_{i=1}^{k-1} \tau_i - \delta, V\right) = \omega_k + \varepsilon. \tag{B2}$$

Define the policy

$$\begin{aligned} P_1 = \big((t_1, \tau_1, \omega_1), \ldots, (t_{k-1}, \tau_{k-1} - \delta, \omega_{k-1} - \varepsilon), \\ (t_{k-1} + f_\rho(A, \omega_{k-1} - \varepsilon), \tau_k + \delta, \omega_k + \varepsilon), (t_{k+1}, \tau_{k+1}, \omega_{k+1}), \ldots, (t_D, \tau_D, \omega_D)\big). \end{aligned}$$

Note that P_1 involves waiting after the k-th dispatch and is feasible for A, so A is not the optimal D-dispatch

area, a contradiction.

Case II: $\omega_{k-1} > \omega_k$ and $f_\rho(A, \omega_k) \geq \tau_k$. Because $t_{k-1} + \tau_k = V$, it follows that $t_{k-1} + f_\rho(A, \omega_k) \geq V$. As $t_{k-1} + f_\rho(A, \omega_{k-1}) + f_\rho(A, \omega_k) = W$, trivially $t_{k-1} + f_\rho(A, \omega_k) + f_\rho(A, \omega_{k-1}) = W$.

Define $\hat{\tau}_{k-1}$ and $\hat{\tau}_k$ such that

$$\Lambda\left(\sum_{i=1}^{k-2} \tau_i, \sum_{i=1}^{k-2} \tau_i + \hat{\tau}_{k-1}\right) = \omega_k \tag{B3}$$

and

$$\Lambda\left(\sum_{i=1}^{k-2} \tau_i + \hat{\tau}_{k-1}, \sum_{i=1}^{k-2} \tau_i + \hat{\tau}_{k-1} + \hat{\tau}_k\right) = \omega_{k-1}. \tag{B4}$$

Note that $\hat{\tau}_{k-1} + \hat{\tau}_k = \tau_{k-1} + \tau_k$. Additionally, $\omega_k < \omega_{k-1}$ implies $\hat{\tau}_{k-1} < \tau_{k-1}$.

Therefore, the policy

$$P_2 = \big((t_1, \tau_1, \omega_1), \ldots, (t_{k-1}, \hat{\tau}_{k-1}, \omega_k), (t_{k-1} + f_\rho(A, \omega_k), \hat{\tau}_k, \omega_{k-1}), (t_{k+1}, \tau_{k+1}, \omega_{k+1}), \ldots, (t_D, \tau_D, \omega_D)\big)$$

is feasible for A. Observe that P_2 was constructed by "swapping" the unscaled dispatch quantities ω_{k-1} and ω_k, then adjusting the accumulation times accordingly. Because $\sum_{i=1}^{k-1} \tau_i = t_{k-1}$ and $\tau_{k-1} > \hat{\tau}_{k-1}$, we have that $\hat{\tau}_{k-1} + \sum_{i=1}^{k-2} \tau_i < t_{k-1}$. This implies that $k_{P_2} < k$, which contradicts the definitions of k and P_0.

Case III: $\omega_{k-1} > \omega_k$ and $f_\rho(A, \omega_k) < \tau_k$. Because f_ρ is continuous and $\lambda(t)$ is non-decreasing, we can find $\tau'_1 < \tau_1, \tau'_2 < \tau_2, \ldots, \tau'_{k-1} < \tau_{k-1}$ such that the following conditions hold.

(i) $f_\rho(A, \omega'_d) = \tau'_{d+1}$ for all $d \in [k-2]$, where ω'_d is defined as

$$\omega'_d = \Lambda\left(\sum_{i=1}^{d-1} \tau'_i, \sum_{i=1}^{d} \tau'_i\right) \tag{B5}$$

for all $d \in [k-1]$.

(ii) Defining

$$\tau'_k = V - \sum_{i=1}^{k-1} \tau'_i \tag{B6}$$

and

$$\omega'_k = \Lambda\left(\sum_{i=1}^{k-1} \tau'_i, V\right), \tag{B7}$$

it holds that $f_\rho(A, \omega'_k) \leq W - V$.

For completeness, define

$$t'_d = \sum_{i=1}^{d} \tau'_d \tag{B8}$$

for all $d \in [k-1]$. Intuitively, this procedure slightly decreases the first $k-1$ accumulation times in such a manner that constraints (B1b) continue to hold at equality for all $d \in [k-1]$, with no waiting after any of the first $k-2$ dispatches. The accumulation time of the k-th dispatch is increased accordingly so that the first k dispatches together together serve all of the accumulated orders in the interval $[0,V]$. The idea is analogous to that illustrated in Figures A5 and A6, except that the non-stationary order rate entails additional notational maneuvering.

To ensure feasibility, it remains to be seen whether the enlarged k-th dispatch can be completed within the interval $[t'_{k-1} + f_\rho(A, \omega'_{k-1}), W]$. Define

$$\delta_\tau = \tau'_k - \tau_k = \sum_{i=1}^{k-1} \tau_i - \sum_{i=1}^{k-1} \tau'_i \tag{B9}$$

and

$$\delta_\omega = \omega'_k - \omega_k = \sum_{i=1}^{k-1} \omega_i - \sum_{i=1}^{k-1} \omega'_i. \tag{B10}$$

Note that δ_τ is the additional accumulation time on the enlarged k-th dispatch, and δ_ω is the additional unscaled quantity on the enlarged k-th dispatch.

Recall that we assumed, without loss of generality, that parameters have been scaled so that $\omega_k = \tau_k$. Because the order rate is non-decreasing, $\delta_\omega \leq \delta_\tau$. Additionally, $f_\rho(A, \omega_k) < \tau_k$ implies $f_\rho(A, \omega_k) < \omega_k$. Because f_ρ is concave and increasing for a fixed area, $f_\rho(A, \omega_k) < \omega_k$ implies $\frac{\partial}{\partial \omega} f_\rho(A, \omega) < 1$ for all $\omega \geq \omega_k$, which in turn implies $f_\rho(A, \omega_k + \delta_\omega) < f_\rho(A, \omega_k) + \delta_\omega$. Therefore,

$$f_\rho(A, \omega'_k) = f_\rho(A, \omega_k + \delta_\omega) < f_\rho(A, \omega_k) + \delta_\tau \leq W - \left(t'_{k-1} + f_\rho(A, \omega'_{k-1})\right), \tag{B11}$$

so the enlarged k-th dispatch can be completed within the desired interval. Formally, the following policy P_3 is feasible to serve A:

$$P_3 = \left((t'_1, \tau'_1, \omega'_1), \ldots, (t'_{k-1}, \tau'_{k-1}, \omega'_{k-1}), (W - f_\rho(A, \omega'_k), \tau'_k, \omega'_k), (t_{k+1}, \tau_{k+1}, \omega_{k+1}), \ldots, (t_D, \tau_D, \omega_D)\right).$$

However, because $f_\rho(A, \omega'_k) < W - \left(t'_{k-1} + f_\rho(A, \omega'_{k-1})\right)$, P_3 involves waiting between the $(k-1)$-th and

k-th dispatches. As such, A cannot be the optimal area, which is a contradiction.

Hence, when $\tau_1, \ldots, \tau_{k-1} > 0$, we have arrived at a contradiction in every case. Suppose instead that one of the first $k-1$ accumulation times for P_0 is zero. If there are $m > 0$ dispatches with zero quantity among the first $k-1$ dispatches (assuming $m \leq D-2$ to avoid triviality), we can remove all m of these dispatches to be left with one of the following:

- some $(D-m)$-dispatch policy feasible for A that involves waiting; this implies $A < \mathcal{A}_{D-m}(\rho) \leq \mathcal{A}^+_{\hat{D}}(\rho)$, a contradiction, or
- some $(D-m)$-dispatch policy feasible for A such that $\sum_{i=1}^{d} \tau_i < t_i$ for some d. Because $\mathcal{A}_D(\rho) = \mathcal{A}^+_{\hat{D}}(\rho)$, we also have $\mathcal{A}_{D-m}(\rho) = \mathcal{A}^+_{\hat{D}}(\rho)$. However, in the previous part of the proof, we showed that $\mathcal{A}_{D-m}(\rho) < \mathcal{A}^+_{\hat{D}}(\rho)$, a contradiction.

Thus, we have shown that $\mathcal{P} = \varnothing$ as desired. □

B.1.1 Counterexample for decreasing order rates

For order rate functions which are not non-decreasing, the policy structure discussed above does not necessarily correspond to maximal zone areas. Consider the following simple counterexample. Let $N = 8$, $T = 10$, $D = D_{\max} = 2$, and $\rho = 0$. Let the dispatch function $f_0(A, \omega) = A\sqrt{\omega}$. The order rate function is $\lambda(t) = 64$ for $t \leq 1$ and $\lambda(t) = 1/7$ for $t > 1$. Figure B1 illustrates this setting.

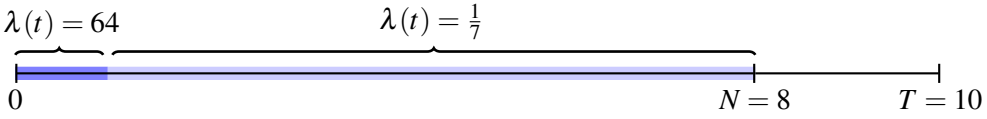


Figure B1: Counterexample problem setting

Observe that the policy $\big((1,1,64),(9,7,1)\big)$ can feasibly serve a zone area of $A = 1$. As illustrated in Figure B2, the first dispatch takes all of the orders placed during the period $[0,1]$, departs at $t_1 = 1$ and returns at $t_2 = 1 + f_0(1,64) = 9$. This leaves the vehicle with just enough time to feasibly complete the second dispatch by the end of the service day.

Suppose we instead seek to adhere to the "optimal" policy structure above, and we wish to serve an area of $A = 1 + \varepsilon$ for any $\varepsilon \geq 0$. The structure dictates that the first dispatch must return at $t = 8$; because the area is greater than 1, it follows that $t_1 = \tau_1 < 0.81$, since the solution to $t + \sqrt{64t} = 8$ is $t = 40 - 16\sqrt{6} < 0.81$.

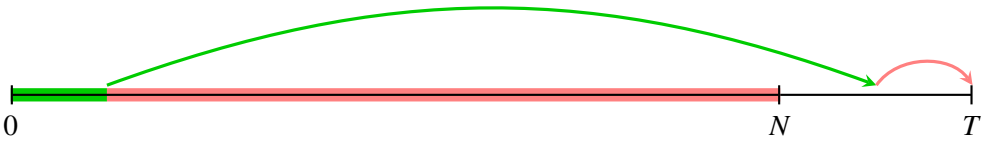


Figure B2: Example dispatching policy with $A = 1$ (to scale)

It follows that $\tau_2 > 7.19$ and $\omega_2 > 7\left(\frac{1}{7}\right) + 0.19(64) = 12.16$. However, $f_0(1+\varepsilon, 12.16) > 2 = T - N$ for any $\varepsilon \geq 0$, so it is impossible for the desired policy structure to induce an optimal area in this setting.

B.2 Calculating maximum zone areas

We return to assuming that $\lambda(t)$ is non-decreasing. The structure of area-maximizing policies in this case is identical to the structure of those in the constant-rate case, so the process of calculating maximum zone areas is quite similar to Algorithm 1. However, it may be impractical to derive closed-form expressions for certain quantities when order rates are non-stationary, so the derivation of the algorithm in this case requires a more general treatment.

For mathematical completeness, define $\lambda(t) = \lambda(N)$ for all $t > N$ and $\lambda(t) = \lambda(0)$ for all $t < 0$. Define the function $h_\rho(A, \tau) = f_\rho\big(A, \Lambda(0, \tau)\big)$, the duration of a dispatch that serves the orders that accumulate over the interval $[0, \tau]$ in a zone with area A. Define $h_\rho^{(0)}(A, \tau) = \tau$ and $h_\rho^{(1)}(A, \tau) = h_\rho(A, \tau)$. For all $d \geq 2$, define the composition

$$h_\rho^{(d)}(A, \tau) = f_\rho\left(A, \Lambda\left(\sum_{k=0}^{d-2} h_\rho^{(k)}(A, \tau), \sum_{k=0}^{d-1} h_\rho^{(k)}(A, \tau)\right)\right). \tag{B12}$$

Informally, $h_\rho^{(d)}(A, \tau)$ is the duration of the d-th dispatch if the first dispatch accumulates orders until τ and all dispatches adhere to conditions (a) and (b) of Theorem 7. Because $\lambda(t)$ is non-decreasing, $h_\rho^{(d)}$ is increasing in both A and τ for all $d \geq 1$. Note that a D-dispatch policy that satisfies all conditions in Theorem 7 must have

$$\sum_{d=0}^{D-1} h_\rho^{(d)}(A, \tau_1) = N \tag{B13}$$

and $h_\rho^{(D)}(A, \tau_1) = T - N$.

For all $A > 0$ and $t \geq 2\rho$, define the function $g_\rho(A, t) = \tau$ such that $f_\rho\big(A, \Lambda(T - t - \tau, T - t)\big) = t$. In other words, $g_\rho(A, t)$ is equal to the unique accumulation time τ such that the time taken to serve the orders which accumulate over the interval $[T - t - \tau, T - t]$ over the area A is equal to t. Define $g_\rho^{(0)}(A, t) = t$ and

$g_\rho^{(1)}(A,t) = g_\rho(A,t)$. For all $d \geq 2$, define the composition $g_\rho^{(d)}(A,t) = \tau$ such that

$$f_\rho\left(A, \Lambda\left(T - \sum_{k=0}^{d-1} g_\rho^{(k)}(A,t) - \tau, T - \sum_{k=0}^{d-1} g_\rho^{(k)}(A,t)\right)\right) = g_\rho^{(d-1)}(A,t). \tag{B14}$$

Informally, $g_\rho^{(d)}(A,t)$ is the accumulation time of the $(D-d+1)$-th dispatch when the zone's area is A and the duration of the D-th (final) dispatch is t, assuming adherence to conditions (a) and (b) of Theorem 7. A D-dispatch policy which satisfies all conditions in Theorem 7 has has $\tau_D = g_\rho^{(1)}(A, T-N)$, $\tau_{D-1} = g_\rho^{(2)}(A, T-N)$, and so on. Because the sum of all accumulation times is N, the area A must satisfy

$$\sum_{d=1}^{D} g_\rho^{(d)}(A, T-N) = N. \tag{B15}$$

The quantity $\sum_{d=1}^{D} g_\rho^{(d)}(A, T-N)$ is continuous and monotonically decreasing in A, so (B15) can have at most one solution. Indeed, when $2D\rho < T$ and $2\rho < T-N$, the area can be chosen such that $\sum_{d=1}^{D} g_\rho^{(d)}(A, T-N)$ is arbitrarily small or arbitrarily large. Thus, for a given D, there is exactly one policy and associated area that satisfies all conditions in Theorem 7, which we can find by numerically solving (B15). To facilitate calculation, we next state and prove the equivalent of Proposition 4 for this case; the statement and proof of Proposition 9 are virtually identical to those of Proposition 4 due to our particular definition of h_ρ and its compositions.

Proposition 9. *Let $\lambda(t)$ be non-decreasing, strictly positive, and integrable. Consider $D \leq D_{\max}(\rho) - 1$. Assuming $\mathcal{A}_D(\rho) = \mathcal{A}_D^+(\rho)$, then $\mathcal{A}_{D+1}(\rho) \geq \mathcal{A}_D^+(\rho)$ if and only if*

$$\sum_{d=1}^{D} h_\rho^{(d)}(\mathcal{A}_D^+(\rho), 0) \leq N. \tag{B16}$$

Proof. Let $\mathcal{A}_D^+(\rho) = \mathcal{A}_D(\rho) > 0$. To first prove the forward implication, suppose that $\mathcal{A}_{D+1}(\rho) \geq \mathcal{A}_D^+(\rho)$. Then, by definition, $\mathcal{A}_{D+1}(\rho) = \mathcal{A}_{D+1}^+(\rho)$; consider the optimal $(D+1)$-dispatch policy, with first departure time $t_1 = \tau_1$, associated with this area. By Theorem 7, the $(D+1)$-th dispatch of this policy departs the depot at time N, so

$$\sum_{d=0}^{D} h_\rho^{(d)}(\mathcal{A}_{D+1}(\rho), t_1) = N.$$

Since $t_1 \geq 0$,

$$\sum_{d=0}^{D} h_\rho^{(d)}(\mathcal{A}_D^+(\rho),0) = \sum_{d=1}^{D} h_\rho^{(d)}(\mathcal{A}_D^+(\rho),0) \leq N$$

because h_ρ and its compositions are component-wise non-decreasing.

To prove the reverse implication, suppose that condition (B16) holds. Consider the optimal D-dispatch policy with first departure time t_1'. By Theorem 7, the D-th dispatch of this policy departs the depot at time N, so

$$\sum_{d=0}^{D-1} h_\rho^{(d)}(\mathcal{A}_D^+(\rho),t_1') = N.$$

Observe that $t_1' \geq 2\rho$ must hold, otherwise

$$\sum_{d=0}^{D-1} h_\rho^{(d)}(\mathcal{A}_D^+(\rho),2\rho) > N,$$

which is a contradiction since

$$\sum_{d=0}^{D-1} h_\rho^{(d)}(\mathcal{A}_D^+(\rho),2\rho) = \sum_{d=0}^{D} h_\rho^{(d)}(\mathcal{A}_D^+(\rho),0) = \sum_{d=1}^{D} h_\rho^{(d)}(\mathcal{A}_D^+(\rho),0) \leq N.$$

Therefore, we can feasibly insert an empty (zero-quantity) dispatch at time 0 in the optimal D-dispatch policy, so $\mathcal{A}_{D+1}(\rho) \geq \mathcal{A}_D^+(\rho)$. □

As a final note, rearranging the dispatch time function gives

$$\mathcal{A}_1(\rho) = \frac{T - N - 2\rho}{\gamma\Lambda(0,N) + \beta\sqrt{\Lambda(0,N)}}. \tag{B17}$$

The full algorithm for calculating maximum zone areas follows. Note that, given the updated definitions of functions f_ρ, g_ρ, and h_ρ, this algorithm is virtually identical to Algorithm 1.

Algorithm B1: Determining maximum zone area with at most $D_{\max}$ dispatches, non-dec. $\lambda(t)$

Input : N, T, ρ, $D_{\max}$, $D_{\max}(\rho)$, and dispatch time function $f_\rho(A,\omega)$

Output: optimal number of dispatches D^* and corresponding area $\mathcal{A}^+_{D_{\max}}(\rho) = \mathcal{A}_{D^*}(\rho)$

initialize $\mathcal{A}_1(\rho) = \frac{T-N-2\rho}{\gamma\Lambda(0,N)+\beta\sqrt{\Lambda(0,N)}}$ and $D^* = 1$

for $D = 2,\ldots,D_{\max}(\rho)$ **do**

  if $\sum_{d=1}^{D-1} h_\rho^{(d)}(\mathcal{A}_{D-1}(\rho),0) \le N$ **then**

    set $\mathcal{A}_D(\rho) \leftarrow A$ s.t. $\sum_{d=1}^{D} g_\rho^{(d)}(A,T-N) = N$

    set $D^* \leftarrow D$

  else

    break

set $\mathcal{A}^+_{D_{\max}}(\rho) \leftarrow \mathcal{A}_{D^*}(\rho)$

Appendix C Computational Details and Data

C.1 Dispatch function details, second computational study

In this section of the appendix, we provide additional computational details on our methodology for fitting dispatch functions for the second computational study. Recall that, for each r in the region, we seek a dispatch function of the form

$$f_\rho(A,\omega) = 2\rho + \alpha + \beta A\sqrt{\omega} + (\beta^{\text{lin}} + \gamma)A\omega,$$

since the number of customers $n = A\omega$, where the unscaled accumulation quantity ω is defined as in Appendix B. In this setting, not only does the linehaul time vary based on the location of r, but the routing constants vary as well due to the inhomogeneity of road speeds in the region. Thus, for each point r, we seek estimates of ρ, β, and β^{lin}. In the remainder of this appendix, we use the notation ρ_r, β_r, and β_r^{lin} to emphasize that these parameters vary with r. Inter-point travel times are calculated with Openrouteservice via VeRoViz and are generally asymmetric.

Our overall approach is to empirically calculate these parameters for a selected number of points, and use interpolation to estimate these parameters for the remaining points in the region. We first examine the linehaul times ρ_r. Let the depot's location be denoted as x_0. We select 50 points $x_1, x_2, \ldots, x_{50}$ in the region located at major road intersections. We first define $\rho_{x_0} = 0$. For all $i \in [50]$, we define ρ_{x_i} such that the total two-way linehaul time $2\rho_{x_i}$ is equal to the total calculated travel time from the depot to x_i and back. For all other points $r \notin \{x_0, x_1, \ldots, x_{50}\}$, we define ρ_r via simple linear interpolation between $\{x_0, x_1, \ldots, x_{50}\}$, with linear extrapolation for points outside conv$(\{x_0, x_1, \ldots, x_{50}\})$. Figure C1 illustrates the resulting value of ρ_r for all points in the region.

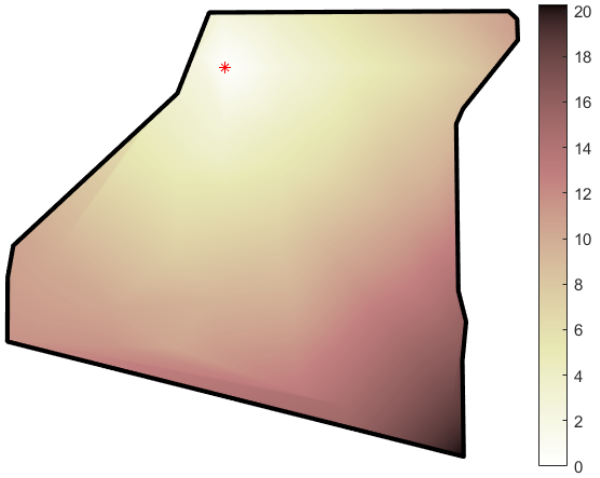


Figure C1: Interpolated values of ρ_r (in minutes)

We next discuss estimating the routing constants β_r and β_r^{lin}. Within the region, we use the 'isochrone' feature in VeRoViz to select ten compact *subregions* $S^1, \ldots, S^{10}$ with corresponding centroids $s_1, \ldots, s_{10}$. Each subregion S^j is defined such that its area $A^j \approx 7$ square miles. The locations and approximate areas

of the subregions were chosen to ensure most of the region is covered by at least one subregion and to demonstrate that a sophisticated methodology is not required for this part of the procedure.

For each S^j and for each of $n \in \{15, 20, 25, \ldots, 75\}$, we randomly generate 50 asymmetric TSP instances of n points in S^j. More specifically, points are generated uniformly at random, then assigned to the closest location on the road network. Ignoring linehaul travel, setup times, and per-order service, each instance is then solved to optimality using Gurobi via Python. For each S^i, we fit the constants β_{s_j}, and $\beta_{s_j}^{\text{lin}}$ via the functional form $\beta_{s_j}\sqrt{A^j n} + \beta_{s_j}^{\text{lin}} n$ on the resulting tour durations. As an example, consider S^1, the subregion in the northwest corner of the region (Figure C2). The empirically estimated corresponding routing constants are $\beta_{s_1} = 3.3921$ and $\beta_{s_1}^{\text{lin}} = 0.8059$. Expressed in minutes, a dispatch function which uses these constants would then have the form

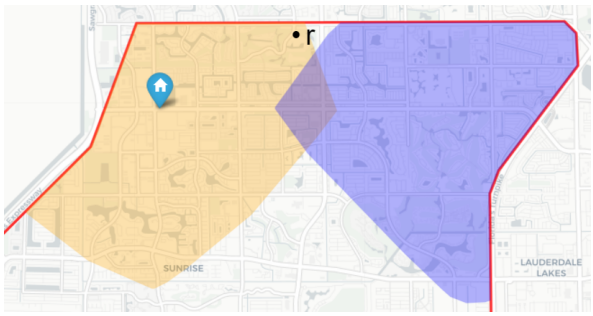


Figure C2: Subregions S^1 (left) and S^3 (right) for routing constant estimation

$$f_\rho(A, \omega) = 2\rho + \alpha + 3.3921A\sqrt{\omega} + (0.8059 + \gamma)A\omega.$$

Finally, we define an interpolation method for estimating β_r and β_r^{lin} at an arbitrary point within the region. For each of the 14 vertices of the region $\{v_1, \ldots, v_{14}\}$, define $\beta_{v_k} = \beta_{s_i^*}$ and $\beta_{v_k}^{\text{lin}} = \beta_{s_i^*}^{\text{lin}}$ where s_i^* is the closest subregion centroid to the vertex v_k. For all other points $r \notin \{s_1, \ldots, s_{10}, v_1, \ldots, v_{14}\}$, we define β_r and β_r^{lin} via simple linear interpolation between $\{s_1, \ldots, s_{10}, v_1, \ldots, v_{14}\}$.

As a complete example, consider the point r denoted in Figure C2. Interpolation for the linehaul time between the depot and r gives $\rho_r = 4.7208$ minutes. Interpolation between the routing constants associated with S^1 and the routing constants associated with S^3 ($\beta_{s_3} = 4.2806$, $\beta_{s_3}^{\text{lin}} = 0.8544$) gives $\beta_r = 3.7581$ and $\beta_r^{\text{lin}} = 0.8259$. Given that $\alpha = 5$ and $\gamma = 2.5$, the resulting dispatch function is

$$f_\rho(A, \omega) = 14.4416 + 3.7581A\sqrt{\omega} + 3.3259A\omega.$$

Suppose $\lambda(t)$ is given by Figure 12. For $D_{\max} = 3$, using this dispatch function and Algorithm B1 to solve for the maximum area of a zone centered at r gives $\mathcal{A}(r) = 6.6412$ square miles.

C.2 Partitioning parameters and data

Table C1: Partitioning input parameters

Study	κ_0	Δ_κ	μ_0	$\sigma_{\max}$	$p_{\max}$	$\varepsilon_{\max}$	$[\mathrm{CLF}_{\min}, \mathrm{CLF}_{\max}]$	δ (mi.)	η
#1	1	0.94	0.3	0.99	300	0.04	$[0.91, 0.99]$	0.1	0.1
#2	1	0.9	0.3	0.98	150	0.04	$[0.89, 0.94]$	0.1	0.1

Study	Θ^-	Θ^+	θ^-	θ^+
#1	0.91	0.99	0.03	0.05
#2	0.88	0.95	0.05	0.05

Table C2: Evolution of the WCVT balancing process, first computational study

Iteration	Centroidal load factor				Max. generator-centroid dist. (mi.)
	Minimum	Average	Maximum	Std. dev.	
1	0.6177	0.9483	1.4663	0.2270	1.6630
10	0.8801	0.9483	1.0698	0.0476	1.1272
20	0.9093	0.9483	0.9772	0.0230	0.5335
30	0.9086	0.9483	0.9819	0.0234	0.1879
40	0.8961	0.9483	0.9894	0.0245	0.1286
50	0.9101	0.9483	0.9856	0.0226	0.0991

Table C3: Evolution of the WCVT balancing process, second computational study

Iteration	Centroidal load factor				Max. generator-centroid dist. (mi.)
	Minimum	Average	Maximum	Std. dev.	
1	0.5889	0.9131	1.3879	0.2614	0.9099
10	0.8572	0.9127	0.9774	0.0394	0.6643
20	0.8861	0.9129	0.9435	0.0203	0.2934
30	0.8598	0.9132	0.9543	0.0303	0.1501
40	0.8815	0.9132	0.9538	0.0221	0.0670
50	0.8141	0.9133	0.9671	0.0438	0.0658
57	0.8955	0.9135	0.9277	0.0089	0.0758

C.3 Detailed simulation results

For each zone, we report the average and standard deviation of the quantity of orders served across the 100 simulated days. This average is compared to the expected number of orders that accumulate in the zone during the pre-cutoff time window used for tactical planning (8 AM to 2 PM in the first study and 8 AM to 4 PM in the second study). The average and standard deviation of the cutoff time is also reported, along with the number of days in which orders were cut off after the cutoff time used for planning.

Table C4: First computational study results

			Orders served			Cutoff (hrs. after 8 AM)		
Zone	CLF	Area (sq. mi.)	Expected	Avg.	Std. dev.	Avg.	Std. dev.	# cutoffs after 2 PM
1	0.9850	14.00	168.00	175.97	4.11	6.19	0.30	79
2	0.9700	12.46	149.52	159.57	5.12	6.41	0.27	91
3	0.9702	11.42	137.04	143.89	4.81	6.36	0.32	85
4	0.9246	9.96	119.50	126.25	5.29	6.38	0.31	88
5	0.9856	13.41	160.92	169.72	5.27	6.30	0.31	82
6	0.9651	12.30	147.60	156.80	6.33	6.48	0.33	91
7	0.9548	10.83	129.96	136.17	5.81	6.24	0.30	77
8	0.9360	9.81	117.72	123.96	6.63	6.43	0.30	93
9	0.9393	9.04	108.50	114.89	5.17	6.38	0.30	86
10	0.9572	11.74	140.88	151.68	5.22	6.46	0.31	90
11	0.9296	10.66	127.92	134.83	6.25	6.36	0.32	89
12	0.9461	9.50	114.05	119.60	5.54	6.32	0.38	78
13	0.9294	8.80	105.55	112.69	5.75	6.43	0.31	90
14	0.9118	8.18	98.13	105.85	6.49	6.49	0.37	94
15	0.9622	11.94	143.28	151.81	5.33	6.46	0.31	90
16	0.9284	10.48	125.78	133.27	6.51	6.33	0.36	81
17	0.9641	10.55	126.60	130.97	5.58	6.28	0.29	86
18	0.9101	9.42	113.05	120.84	5.79	6.45	0.33	88
Avg.	0.9483	10.81	129.67	137.15	5.61	6.37	0.32	86.56
Total		194.50	2334	2468.76	21.73			

Table C5: Second computational study results

Zone	CLF	Area (sq. mi)	Orders served			Cutoff (hrs. after 8 AM)		# cutoffs after 4 PM
			Expected	Avg.	Std. Dev.	Avg.	Std. Dev.	
1	0.9119	6.33	101.35	106.90	5.42	8.27	0.36	73
2	0.9159	5.74	91.78	94.77	5.56	8.20	0.38	72
3	0.9277	6.08	97.27	100.89	5.40	8.21	0.34	72
4	0.8955	5.67	90.72	97.05	6.43	8.38	0.38	86
5	0.9124	5.60	89.57	92.88	5.88	8.20	0.41	70
6	0.9092	5.37	85.85	88.55	5.77	8.17	0.40	69
7	0.9128	5.52	88.39	90.45	5.79	8.13	0.41	64
8	0.9228	5.49	87.78	91.60	6.05	8.24	0.35	77
Avg.	0.9135	5.72	91.59	95.39	5.79	8.22	0.38	72.88
Total		45.79	732.71	763.09	14.97			

C.4 *A posteriori* Formulation

We use the following integer programming formulation to compute *a posteriori* bounds on the number of orders served over the course of a service day.

Parameters:

- Node set: order locations $[L]$, depot 0
- Edge set: unordered pairs of nodes $E = \big\{\{i, j\} \mid 0 \leq i < j \leq L\big\}$
- Release times (order placement times): r_i for all $i \in [L]$, with $r_1 < r_2 < \ldots < r_L$
- Travel times: c_e for all $e \in E$; these times do not include service or setup times
- Number of dispatches D, deadline T, setup time at depot α, and per-order service time γ

Variables:

- $x_e^d = 1$ if the edge $e \in E$ is used on the d-th dispatch, 0 otherwise
- $y_i^d = 1$ if order i is on the d-th dispatch, 0 otherwise
- t_d, the dispatch departure time for each $d \in [D]$

Model:

$$\max_{t,x,y} \quad \sum_{d=1}^{D} \sum_{i=1}^{L} y_i^d$$

$$\text{s.t.} \quad \sum_{e \in \delta(0)} x_e^d = 2 \qquad \forall d \in [D], \tag{C1}$$

$$\sum_{e \in \delta(i)} x_e^d = 2y_i^d \qquad \forall i \in [L], d \in [D], \tag{C2}$$

$$\sum_{d=1}^{D} y_1^d \leq 1, \tag{C3}$$

$$\sum_{d=1}^{D} y_{i+1}^d \leq \sum_{d=1}^{D} y_i^d \qquad \forall i \in [L], \tag{C4}$$

$$t_d + \sum_{e \in E} (c_e + \gamma) x_e^d - \gamma + \alpha \leq t_{d+1} \qquad \forall d \in [D-1], \tag{C5}$$

$$t_D + \sum_{e \in E} (c_e + \gamma) x_e^D - \gamma + \alpha \leq T, \tag{C6}$$

$$t_d \geq r_i y_i^d \qquad \forall i \in [L], d \in [D], \tag{C7}$$

$$\sum_{e \subseteq S} x_e^d \leq |S| - 1 \qquad \forall \varnothing \neq S \subseteq [L], d \in [D], \tag{C8}$$

$$x_e^d \in \{0,1\} \qquad \forall e \in E, d \in [D], \tag{C9}$$

$$y_i^d \in \{0,1\} \qquad \forall i \in [L], d \in [D]. \tag{C10}$$

The objective is to maximize the number of orders served. Constraints (C1) ensure that the depot is on every tour. Constraints (C2) ensure that a node is on tour d if and only if it is served by the d-th dispatch. Constraints (C3) limit the first order (and implicitly, all future orders) to be on at most one dispatch. Constraints (C4) allow order $i+1$ to be served only if order i is served. Constraints (C5) and (C6) ensure that dispatch return times are feasible with respect to the next dispatch departure time or the deadline. Constraints (C7) ensure that dispatch departures occur only after all associated orders have been placed. Subtour elimination constraints (C8) are added during the optimization as necessary.

The operational solution is used as a warm start to the model. The optimization is initially run with a time limit of one minute. The resulting upper bound is used as the value of L for the final optimization with a time limit of 30 minutes.