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

December 28, 2020

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. 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% [52]. 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 [1]. 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. [2], and traditional retail stores such as Target and Walmart have also begun offering SDD services in recent years [45].

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 [26]. The result is that SDD systems must often dispatch individual vehicles on multiple routes during the service day, serving fewer customers per time. Controlling costs in such low-density 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., 15, 17, 24, 54]. 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 [40], 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. Operations 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 [44]. 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). Using continuous approximations, [44] study cost-minimizing dispatching policies for a single vehicle operating in a service region making multiple dispatches per day, and for a fleet of vehicles where each serves customers scattered throughout the service region. Approximation results are then used to analyze selected tactical design problems. Missing in this analysis are results for a fleet of vehicles that serve sub-regions within the larger service region. For settings with larger service
regions that may require many hundreds or thousands of deliveries per day, there can be significant performance benefits from geographically grouping or partitioning customers into zones dedicated to individual vehicles [e.g., 8]. Motivated by these considerations, this paper considers design problems for SDD systems where it is impractical to have vehicles serve a distribution center’s entire service region.

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 proofs, algorithms, 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., 47, 48] provide useful models for the random nature of orders placed within a SDD system; [35, 36] survey models and solution methods for the broader class of stochastic VRPs. The pickup and delivery VRP [41] and the VRP with customer release times [13] 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 [14, 37]. 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 [38, 39].

An example of such an operational problem is the same-day delivery problem for online purchases (SDDP-OP). As introduced by [54], the setting of the SDDP-OP involves a homogeneous fleet of fixed size and a single depot. The objective is to maximize the expected total number of fulfilled orders. The problem is modeled as a Markov decision process and solved via approximate dynamic programming. In [15], the authors study an extension to the SDDP-OP in which vehicles are allowed to pre-emptively return to the depot to pick up additional orders before completing a route. A secondary objective of minimizing total routing cost is also included. Another operational SDD model is the dynamic dispatch waves problem (DDWP) as studied in [24, 25, 26]. 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 penalty costs for unserved orders. In [25], the authors consider the DDWP in one dimension; the deterministic version of the problem is solved using a dynamic programming approach, and heuristics are proposed for the stochastic version. The two-dimensional DDWP is studied in [24]. In [26], a variant is studied in which order requests are immediately accepted or rejected upon arrival.

Additional operational SDD problems studied in the literature consider drones [17, 50], autonomous vehicles [49], and other extensions [46, 51, 56]. 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 [46]), doing so...
can be computationally expensive. Additionally, operational problems are often solved heuristically without optimality guarantees, potentially decreasing the reliability of any associated insights. The authors in [44], 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 [5] 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 $\gamma \sqrt{An}$ as $n$ increases, where $\gamma$ 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 [16] and with different underlying metrics [29, 30]. Empirical estimates of the BHH routing constant for varying tour sizes are calculated by [4, 23] for Euclidean and rectilinear (Manhattan) distances, respectively, and [28] 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 [6, 12, 53]. Of particular relevance to our work is [20], in which continuous approximations are used in conjunction with mixed-integer linear programming to solve a fleet sizing and composition problem for rectangular service regions. The authors assume that all delivery requests are known in advance, while we assume requests arrive dynamically over the course of the service day. Comprehensive surveys of logistical applications of continuous approximation methods are presented in [3, 21].

In [44], 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 $n$ orders, where $n$ is not necessarily integral, is represented by a concave, increasing routing 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 model average-case system behavior in order to reveal tactical insights. Vehicles dispatch from a single depot, and each vehicle serves orders from the entire region (i.e., no partitioning of the service 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 routing time. Results are developed both for single-vehicle fleets and for fleets where most vehicles are dispatched once during the operating day, and characterizations of the optimal (cost-minimizing) dispatching policies are given for each case. For the optimal single-vehicle policy, it is shown under additional assumptions that the vehicle always takes all unserved orders at the time
of each dispatch and never waits at the depot after its first dispatch. It is also shown how the model and the associated optimal policies can be used to choose order cutoff times and to manage orders that have accumulated at the start of the service day. Other detailed tactical design issues, including fleet sizing for partitioned service regions, are left as avenues for further study.

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 in which each vehicle is independently responsible for serving demand within a zone. As discussed by [57], such a scheme improves efficiency by increasing drivers’ familiarity with a smaller geographical area; we discuss potential extensions and modifications to this scheme in Section 6. 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. This underlying concept can be generalized into higher dimensions or via arbitrary metrics; see [32] for examples of extensions and applications of Voronoi tessellations across a range of fields. 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 in logistics, including for facility location [e.g., 11, 31, 34] and the design of vehicle routing zones [10, 22, 33]. A centroidal Voronoi tessellation is another extension in which the generator and centroid of each zone coincide [18]. We propose an approximate two-stage weighted centroidal Voronoi tessellation (WCVT) scheme for service region partitioning adapted from [22, 33, 34], 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. While less suitable for our problem setting, other partitioning schemes for vehicle routing zones include those that incorporate large neighborhood search heuristics [27] and iterative ham-sandwich cuts [9].

3. **Area Maximization and Fleet Sizing**

We consider an SDD system operating in a simply connected, two-dimensional service region \( R \) with a single depot from which all vehicles are dispatched. The system is characterized as follows:

- Demand for same-day deliveries (i.e., orders) accumulates continuously throughout the service region at a rate of \( \lambda \) orders per unit area per unit time. Orders accumulate from the beginning of the service
day \( t = 0 \) 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 serving (i.e., delivering) orders that accumulate in a distinct zone within the region. The fleet is assumed to be homogeneous.
- 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 \( D_{\text{max}} \) times over the course of the service day, where the parameter \( D_{\text{max}} \) is selected by the system designer. Individual vehicles may operate different dispatch schedules. Vehicles are uncapacitated and not constrained to carry an integer number of orders.

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 served, and by the tendency for delivery vehicles to operate under full capacity when time constraints significantly restrict flexibility [24, 25, 44].

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 \( \alpha \), with \( \alpha \geq 0 \);
(ii) a total linehaul time \( 2\rho \), representing the vehicle’s travel time to and from the zone;
(iii) a service time \( \beta n \) proportional to the number of orders, where \( \beta \geq 0 \) is the service time per order;
(iv) a BHH routing time between orders \( \gamma \sqrt{An} \), where \( \gamma > 0 \) is an appropriately chosen BHH constant.

The use of the BHH routing time to create a good approximation of travel time within the zone relies on the assumption that order locations are independently sampled from a continuous distribution over the zone [44] and that the shape of the zone is sufficiently “compact” in the general (i.e., non-topological) sense. 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 $\tau$ is $A\tau$. Therefore, the total time a vehicle takes to serve the orders accumulating over a period of length $\tau$ with a single dispatch is given by the dispatch time function $f_\rho(A, \tau) = 2\rho + \alpha + \beta A \tau + \gamma A \sqrt{\tau}$. The dispatch time function $f_\rho$ is continuous, component-wise increasing, and non-negative for non-negative $\tau$ and $A$. Additionally, for fixed $A > 0$, $f_\rho$ is strictly concave in $\tau$.

To solve the fleet sizing problem that determines the minimum number of vehicles required to service the region, we first consider the related single-vehicle zone area maximization problem: given a fixed centroid with associated linehaul travel time $\rho$, we seek to determine the maximum area of a zone around the centroid, independent of other zones and service region boundaries, such that the vehicle can feasibly serve all orders with at most $D_{\text{max}}$ dispatches. Then, we show how an approximation of the minimum fleet size follows directly from a characterization of the relationship between $\rho$ and the associated maximum zone area.

### 3.1 Zone area maximization

Consider a single-vehicle zone, associated dispatch time function $f_\rho$, 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 $\rho$).

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 $((t_1, \tau_1), (t_2, \tau_2), \ldots, (t_D, \tau_D))$. The dispatch departure times $t_1, \ldots, t_D$ satisfy $0 \leq t_1 \leq \ldots \leq t_D \leq T$. For each $d \in [D]$, the corresponding $\tau_d$ represents the duration of time during which orders accumulate to be served by the $d$-th dispatch. We refer to $\tau_d$ 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
Figure 2: Dispatching policy example

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

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 \(t = 15\) with all of the orders accumulated between \(t = 0\) and \(t = 15\), departs again at \(t = 40\) with all of the orders that accumulated between \(t = 15\) and \(t = 35\), and departs again with all of the remaining orders at \(t = 70\). The policy would be denoted as \(((15, 15), (40, 20), (70, 30))\).

Figure 2 illustrates this dispatching policy to scale (with arrows representing the dispatch departure and arrival times) when \(A = 1\) and \(f_{\rho}(A, \tau) = 8 + 0.1A\tau + 2A\sqrt{\tau}\).

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

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 \(2\rho\), 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 \(A_D(\rho) = 0\). Observe that, for a given value of \(\rho\), it may not be feasible for the vehicle to complete \(D_{\text{max}}\) dispatches. As such, for all \(\rho\), define \(D_{\text{max}}(\rho) \leq D_{\text{max}}\) 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 \(0 < D \leq D_{\text{max}}(\rho)\), we define \(A_D(\rho) \geq 0\) as the optimal value of the following non-convex optimization problem:
\[
A_D(\rho) = \max_{t, \tau_d} A
\]

s.t. \( t_d \geq \sum_{i=1}^{d} \tau_i \) \( \forall d \in [D] \), \hfill (1b)

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

\( T \geq t_D + f_\rho(A, \tau_D) \), \hfill (1d)

\[ \sum_{d=1}^{D} \tau_d = N \], \hfill (1e)

\( \tau_d \geq 0 \) \( \forall d \in [D] \). \hfill (1f)

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 (1c) 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 \( D = 1 \), 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 \( 2\rho + \beta AN + \gamma A \sqrt{N} = T - N \). Solving the equation for \( A \) gives

\[
A_1(\rho) = \frac{T - N - 2\rho}{\beta N + \gamma \sqrt{N}}.
\]

However, model (1) does not readily admit a closed-form optimal solution for values of \( D \) greater than one. To develop a procedure for calculating \( A_D(\rho) \) 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 $t_D + f_p(A, \tau_D) < T$. For some area $A$, we say that an associated policy $((t_1, \tau_1), \ldots, (t_D, \tau_D))$ 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 $P = ((t_1, \tau_1), \ldots, (t_D, \tau_D))$. If $P$ involves waiting, there exists a feasible policy $\hat{P} = ((\hat{t}_1, \hat{\tau}_1), \ldots, (\hat{t}_D, \hat{\tau}_D))$ 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 $D \leq D_{\text{max}}(\rho)$. Any optimal $D$-dispatch policy for $A_D(\rho)$ does not involve waiting.

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

Therefore, when $A = A_D(\rho)$, constraints (1c) and (1d) hold at equality. This result is consistent with other SDD contexts [25, 44], 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 associated 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 $D_{\text{max}}$ dispatches are allowed.

**Theorem 3.** Let $D \leq D_{\text{max}}(\rho)$. If $A_D(\rho) = \hat{A}_{D_{\text{max}}}(\rho)$, 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 $A = A_D(\rho) = \hat{A}_{D_{\text{max}}}(\rho)$, constraints (1b) hold at equality.
Figure 3: No-wait dispatching policy satisfying conditions (a) and (b) of Theorem 3

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 $t \geq 2\rho$ 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 $f_\rho(A,\tau) = 2\rho + \gamma A\sqrt{\tau}$,

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

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

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

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

Define $f_\rho^{(0)}(A,\tau) = \tau$ and $f_\rho^{(k)}(A,\tau) = f_\rho(A,f_\rho^{(k-1)}(A,\tau))$ for all $k \geq 1$; $f_\rho^{(k)}$ is the duration of the $k$-th dispatch when the first occurs at time (and accumulates) $\tau$. Similarly, define $g_\rho^{(0)}(A,t) = t$ and $g_\rho^{(k)}(A,t) = g_\rho(A,g_\rho^{(k-1)}(A,t))$ for all $k \geq 1$; here, $g_\rho^{(k)}$ is the accumulation time for the $(D-k+1)$-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

$$\sum_{d=1}^{D} g_\rho^{(d)}(A,T-N) = N.$$  (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 $2DP < T$ and $2P < T - N$, 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 the conditions in Theorems 2 and 3.

Suppose now that \( A_D(\rho) = \hat{A}_D(\rho) \); 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 \( \hat{A}_D(\rho) \) can be feasibly served with exactly \( D + 1 \) dispatches.

**Theorem 4.** Consider \( D \leq D_{\max}(\rho) - 1 \). If \( A_D(\rho) = \hat{A}_D(\rho) \), then \( A_{D+1}(\rho) \geq \hat{A}_D(\rho) \) if and only if

\[
\sum_{d=1}^{D} f^{(d)}_\rho (\hat{A}_D(\rho), 0) \leq N. \tag{5}
\]

**Proof.** Let \( \hat{A}_D(\rho) = A_D(\rho) > 0 \). To first prove the forward implication, suppose that \( A_{D+1}(\rho) \geq \hat{A}_D(\rho) \). Then, by definition, \( A_{D+1}(\rho) = \hat{A}_{D+1}(\rho) \); 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

\[
\sum_{d=0}^{D} f^{(d)}_\rho (A_{D+1}(\rho), t_1) = N.
\]

Since \( t_1 \geq 0 \),

\[
\sum_{d=0}^{D} f^{(d)}_\rho (\hat{A}_D(\rho), 0) = \sum_{d=1}^{D} f^{(d)}_\rho (\hat{A}_D(\rho), 0) \leq N
\]

because \( f_\rho \) and its compositions are component-wise increasing.

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

\[
\sum_{d=0}^{D-1} f^{(d)}_\rho (\hat{A}_D(\rho), t'_1) = N.
\]

Observe that \( t'_1 \geq 2\rho \) must hold, otherwise

\[
\sum_{d=0}^{D-1} f^{(d)}_\rho (\hat{A}_D(\rho), 2\rho) > N,
\]

which is a contradiction since

\[
\sum_{d=0}^{D-1} f^{(d)}_\rho (\hat{A}_D(\rho), 2\rho) = \sum_{d=0}^{D} f^{(d)}_\rho (\hat{A}_D(\rho), 0) = \sum_{d=1}^{D} f^{(d)}_\rho (\hat{A}_D(\rho), 0) \leq N.
\]
Therefore, we can feasibly insert an empty (zero-quantity) dispatch at time 0 in the optimal $D$-dispatch policy, so $A_{D+1}(\rho) \geq \hat{A}_D(\rho)$.

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

$$\sum_{d=1}^{D'} f^{(d)}_\rho (\hat{A}_D(\rho), 0) \geq \sum_{d=1}^D f^{(d)}_\rho (\hat{A}_D(\rho), 0) > N.$$ 

It follows that, if $\hat{A}_D(\rho) = A_D(\rho)$ and $A_{D+1}(\rho) < \hat{A}_D(\rho)$, then $A_{D'}(\rho) < \hat{A}_D(\rho)$ for all $D' > D$. We may therefore conclude that, for a given $\rho$, the optimal areas are non-decreasing in the number of dispatches until some optimal $D^*$, after which $A_{D'}(\rho) > A_{D'}(\rho)$ for all $D' \in \{D^* + 1, \ldots, D_{\text{max}}\}$. Thus, the problem of calculating $\hat{A}_{D_{\text{max}}}(\rho)$ is equivalent to finding $D^*$ and the corresponding area $A_{D^*}(\rho)$.

Algorithm 1 consolidates our results into an iterative root-finding procedure for solving the zone area maximization problem. For a given $\rho$ and maximum number of dispatches, the algorithm calculates $\hat{A}_{D_{\text{max}}}(\rho)$ by determining $D^*$ and the corresponding area $A_{D^*}(\rho)$. The specific method used to solve $\sum_{d=1}^D g^{(d)}_\rho (A, T - N) = N$ should be chosen with parameters appropriately tuned to ensure numerical stability. We use a combination of Scipy.optimize.fsolve and simple local search when an estimate of $A_d(\rho)$ is available (e.g., from a previously found $A_d(\rho \pm \epsilon)$) and Scipy.optimize.brenth otherwise.

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

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

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

1. initialize $A_1(\rho) = \frac{T-N-2\rho}{\beta N + \gamma \sqrt{N}}$ and $D^* = 1$

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

3.   if $\sum_{d=1}^{D-1} g^{(d)}_\rho (A_{D-1}(\rho), 0) \leq N$ then

4.     set $A_D(\rho) \leftarrow A \text{ s.t. } \sum_{d=1}^D g^{(d)}_\rho (A, T - N) = N$

5.     set $D^* \leftarrow D$

6.   else

7.     break

8. end

9. end

10. set $\hat{A}_{D_{\text{max}}}(\rho) \leftarrow A_{D^*}(\rho)$

In order to obtain an absolute limit on the area a vehicle can serve under any circumstances, we consider
the limiting behavior of $\hat{A}_{D_{\max}}(\rho)$ as $D_{\max} \to \infty$, and whether this limit is bounded. Consider the limit
\[ \lim_{D \to \infty} A_D(\rho); \] 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 $D$ maximizes $A_D(\rho)$, and this quantity is bounded above by $\hat{A}_D(0)$. When $\rho = \alpha = 0$, the $D$-dispatch policy associated with $A_D(0)$ is also feasible for $A_{D+1}(0)$ if an additional dispatch with zero quantity is appended. Hence, $A_1(0) \leq A_2(0) \leq A_3(0) \leq \cdots$ by induction, and $A_\infty = \lim_{D \to \infty} A_D(0) = \lim_{D_{\max} \to \infty} \hat{A}_{D_{\max}}(0)$ is nonzero.

It can be easily seen that $A_\infty$ is finite. For any number of dispatches $D$ and area $A$, a lower bound for the total dispatch time across all dispatches is $\beta AN + \gamma A\sqrt{N}$ by the concavity of $f_\rho$ for fixed $A$. Because all dispatches must take place in the time interval $[0,T]$, it follows that $A_D(\rho) \leq \frac{T}{\beta N + \gamma \sqrt{N}}$ for any $\rho$ and $D$. Therefore, $A_\infty$ is bounded above by this finite quantity as well. By our previous results, $A_\infty$ is the unique area that satisfies
\[ \sum_{d=1}^{\infty} g_0^{(d)}(A_\infty, T - N) = N. \] (6)
While this series likely has no closed form (even when $\beta = 0$), we empirically observe that the terms of the series rapidly approach zero, and $A_\infty$ 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. It follows that, when $\rho > 0$, the limit $\lim_{D_{\max} \to \infty} \hat{A}_{D_{\max}}(\rho) = \hat{A}_D(\rho)$ is similarly finite and bounded above by $\hat{A}_D(0) < A_\infty$.

Figure 4 presents optimal area functions for two different parameter settings with $T = 100$ for all $\rho$ such that $A_1(\rho) > 0$. For both cases, for each $D \leq 4$, the function $A_D(\rho)$ is plotted for all $\rho$ where $A_D(\rho) = \hat{A}_D(\rho)$. 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 $\frac{\rho}{2}$. In the first case with $\alpha = 0$, we estimate $A_\infty$ by approximating the infinite series in (6) with its first ten terms. We observe well-behaved, although not necessarily linear, area functions that facilitate precise polynomial approximations. The optimal area functions also exhibit diminishing returns as more dispatches are added. When the functions $A_D(\rho)$ and $A_{D+1}(\rho)$ intersect, they do so at the point $(\rho,A)$ 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 $N = 60$, $A_2(\rho) > A_1(\rho)$ for all $\rho$. In the second case, where $N = 70$, $A_3(\rho) > A_2(\rho) > A_1(\rho)$ for all $\rho$. We close our discussion of optimal area functions with two related results that generalize this observation. The
Figure 4: Optimal area function examples

\[ f_\rho(A, \tau) = 2\rho + 0.3A\tau + 1.2A\sqrt{\tau}, \quad N = 60, \quad T = 100 \]

\[ f_\rho(A, \tau) = 2\rho + 0.5 + 0.1A + 0.5A, \quad N = 70, \quad T = 100 \]

former follows directly from Theorems 2 and 3 and the fact that \( f_\rho \) is concave and increasing for fixed \( A \). The latter is proved here for \( D = 2 \) and proved for \( D \geq 3 \) in Appendix A.

Lemma 5. Let \( 2 \leq D \leq D_{\text{max}}(\rho) \), and suppose \( A_D(\rho) = \hat{A}_{D_{\text{max}}}(\rho) \).

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

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

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

Theorem 6. For any \( D \geq 2 \), \( \frac{N}{T} \geq \frac{D-1}{D} \) if and only if \( A_D(\rho) > A_{D-1}(\rho) > \cdots > A_1(\rho) \) for all \( \rho \) satisfying \( 2\rho < T - N \).

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 \( A_1(\rho) \) is associated with a feasible dispatching policy \( ((t_1, \tau_1)) \) 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, \( (0, 0), (t_1, \tau_1) \) is a feasible two-dispatch policy for the area \( A_1(\rho) \) that includes waiting after the first dispatch. Therefore, \( A_2(\rho) > A_1(\rho) \). 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 \( A_2(\rho^*) = 0 < A_1(\rho^*) \).
3.4 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 $D_{\text{max}}$ dispatches over the course of the service day. For all points $r \in R$, let $\rho_r$ denote the travel time from the depot to $r$. Because the maximum total area of a zone centered at $r$ is $\hat{A}_{D_{\text{max}}} (\rho_r)$, we require at least $1/\hat{A}_{D_{\text{max}}} (\rho_r)$ vehicles to serve a subregion 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 [19, p. 154]:

$$V_{CA} = \int_{r \in R} \frac{1}{\hat{A}_{D_{\text{max}}} (\rho_r)} \, dr. \quad (7)$$

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 will lead to an average zone size less than $\hat{A}_{D_{\text{max}}} (\rho)$ in the resulting partition, increasing the overall robustness of the system. On the other hand, using fewer than $V_{CA}$ vehicles will lead to an average zone size greater than $\hat{A}_{D_{\text{max}}} (\rho)$ 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.

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 [22, 33].

For notational convenience, define the function $A(\cdot)$ for all points $r \in R$ as $A(r) = \hat{A}_{D_{\text{max}}} (\rho_r)$. 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 $\text{area}(Z)/A(\text{centroid}(Z))$. Our goal is to approximately balance the zone CLFs across $R$, as measured by all CLFs falling within a user-defined interval (e.g., $[0.9, 1]$) or by the range of all CLFs being less than a user-defined value (e.g., 0.1).
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 balancing loads.

4.1 Initial generator locations
To select initial generator locations, we combine our maximum area calculations with an “automated sliding procedure” similar to that proposed in [33, 34]. We present a high-level overview of our version of the method here and defer additional examples and further discussion of the implementation details 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 $A(c_i)$. 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

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

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 [34], 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 $C_i \setminus R \neq \emptyset$, 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 $\mu$, and areas are updated based on the new locations of the disk centers. As recommended by [34], we decrease the step size by a multiplicative factor of $\Delta \mu$ 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 below some user-defined tolerance and no disks extend outside of the region.

As in [34], we additionally include a shrinkage coefficient \( \kappa \) by which all disk areas are scaled; \( i.e. \), the area of a disk at a point \( r \) is \( \kappa \cdot A(r) \). This ensures the eventual termination of the procedure, since disk overlaps vanish if \( \kappa \) is sufficiently small. If we do not find a solution in \( p_{\text{max}} \) iterations, we reduce \( \kappa \) by performing the update \( \kappa \leftarrow \kappa \cdot \Delta \kappa \) (where \( 0 \ll \Delta \kappa < 1 \)), reset \( \mu \) to its initial value \( \mu_0 \), and continue. The initial shrinkage coefficient \( \kappa_0 \) should depend on the ratio of \( m \) to \( V_{\text{CA}} \). If \( m > V_{\text{CA}} \), using \( \kappa_0 = 1 \) is sufficient; otherwise, a value of \( \kappa_0 \) larger than \( V_{\text{CA}} / m \) 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” [32] or “degenerate locations” [34]). We add a small random perturbing force to each disk at each iteration to prevent this occurrence. The detailed method is included in Appendix B.

The resulting configuration of disk centers exhibit spacing between each other and with the boundaries of the region. The final value of \( \kappa \) represents the factor by which all zones have been scaled down relative to their maximum area. Figures 7a and 12a in Section 5 show example outputs of this sliding method. The disk centers can be manually adjusted further if a partitioning structure is evident, as in our first computational study, or directly used as initial generator locations for the WCVT balancing procedure, as in our second computational study.

### 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 balanced with respect to our criteria. In order to improve the solution, we employ an iterative WCVT load balancing procedure [22, 33] under the assumption that the service region
has been discretized into small blocks, which is likely the situation faced by an urban SDD system manager.

The inputs to the procedure are blocks $B_1, \ldots, B_\ell$ with corresponding centroids $b_1, \ldots, b_\ell$, initial generator locations $c_1, \ldots, c_m$, and initial weights $(w_1, \ldots, w_m) = (1, \ldots, 1)$. 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 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 $\eta$, 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 stopping condition 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. Possible additional stopping criteria include some combination of:

(i) all zone CLFs falling within a user-defined interval;

(ii) the range of zone CLFs being less than some value;

(iii) the maximum distance between any zone’s generator and centroid being less than some value; or

(iv) the generators’ movement between consecutive iterations being less than some tolerance $\theta$.

The two former criteria measure balance across zones, while the two latter criteria signal 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 $[0.9, 0.97]$ and $[0.91, 0.99]$, respectively, to observe the effects of CLF on measures of service level. To this end, we include lower and upper bounds $\Theta^-$ and $\Theta^+$ to speed up convergence of CLFs into the desired interval. For a zone $Z_i$ with generator $z_i$, if $\text{area}(Z_i) / A(z_i) > \Theta^+$, the zone’s weight is increased by $\theta^+$, and if $\text{area}(Z_i) / A(z_i) < \Theta^-$, the weight is decreased by $\theta^-$, where $0 \leq \theta^- < \Theta^+ \leq 1$, $0 < \theta^- \ll 1$, $0 < \theta^+ \ll 1$. We also use condition (iii) with maximum distances of 0.05 miles and 0.1 miles, respectively. The details of the full procedure are included in Appendix B.
5. **Computational Studies**

We empirically test our approach in an operational setting to evaluate the model’s approximations and predictions. We simulate an SDD operational setting in which orders arrive according to a Poisson point process, and where exact vehicle routing times are computed based on order delivery locations modeled as points in $\mathbb{R}^2$. Because our primary motivation is the design of SDD in urban settings, we model driving distances and times using a Manhattan (rectilinear) distance metric in these studies. In our first study setting, we consider a regular, diamond-shaped service region with a depot located in the center and dense order arrivals. In the second, we consider an irregularly-shaped service region with a depot located in a corner and sparser order arrivals. While the cutoff time is fixed when estimating fleet size and building a zone partition design, we allow for dynamic cutoff time adjustments in our operational simulations which reflects how some SDD systems are managed in practice [44]. We consider the tactical cutoff time to be a target (possibly representing the cutoff time advertised to customers), and compare the simulated operational cutoff times to this target. We use approximately $1.05 V_{CA}$ vehicles as a slightly conservative fleet size estimate that increases the system’s robustness, making it more likely that orders can be placed up to (and beyond) the cutoff time in each zone and still be delivered on time.

We compute the fleet sizing integrals numerically in Python 3.7.3 by the method of [55], as implemented in the quadpy library [42], on polygonal service regions triangulated with the tripy library [7]. We create the service region partitions in MATLAB R2019b, and we calculate optimal TSP solutions using an arc-based integer programming formulation implemented in Gurobi 9.0.1 via Python.

To develop dispatch time functions for the tactical model, we require a suitable BHH routing constant $\gamma$, 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 [5]. The rectilinear BHH constant is empirically estimated as 0.8943 by [23] for large $n$. However, the best $\gamma$ constant is asymptotically decreasing in the number of stops, and thus it is helpful to build an estimate more appropriate for scenarios 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 in our dispatch functions, corresponding to our estimate for $n = 30$. 

21
5.1 Depot located in center

In our first study, we consider an 8 \times 8 mile service region, oriented as a diamond, with a depot located in the center. We use rectilinear distances and assume a vehicle speed of 25 miles per hour, a service time per order of two minutes, and a setup time per dispatch of five minutes. The service day extends from 8 AM to 8 PM with a target order cutoff time of 3:30 PM, and each vehicle is allowed at most \( D_{\text{max}} = 3 \) dispatches per day. Orders arrive at a rate of 10 per hour per square mile. To ensure that the order arrival rate is \( \lambda = 1 \), we re-scale all parameters to be measured in increments of six minutes.

Figure 6 displays the resulting dispatch time function, re-scaled time parameters, and optimal area functions for up to five dispatches per vehicle. Within the region, the scaled linehaul time \( \rho \) ranges from 0 to 2.2627. The continuous approximation fleet size guideline is \( V_{CA} = 21.65 \) vehicles for \( D_{\text{max}} = 3 \). For comparison, the fleet size guidelines are 44.58, 24.93, 20.85, and 20.75 for \( D_{\text{max}} = 1, 2, 4, \) and 5, respectively. We partition the service region into \( 1.05V_{CA} = 22.73 \approx 23 \) single-vehicle zones. We contrast these results with the “many-vehicle” policy described by [44], which does not partition the service region and requires a much larger fleet of 43 vehicles. This result illustrates the value of \textit{a priori} service region partitioning for large-scale SDD systems.

\[ f_{\rho}(A, \tau) = 2 \rho + 0.8333 + 0.3333A \tau + 0.4213A\sqrt{\tau}, \quad N = 75, \quad T = 120 \]

To begin constructing the 23 vehicle zones, the automated sliding procedure terminated after 2,784 total iterations in just over 43 seconds of runtime with a final shrinkage coefficient of \( \kappa = 0.8216 \). The resulting output, shown in Figure 7a, suggests a partitioning structure of three central diagonal “rows” of five zones each between two outer rows of four zones each. The chosen initial WCVT generator locations, shown in Figure 7b, are identical to this output except that the generators in the two outer rows of four have been evenly spaced apart.

To construct the partition, we discretized the service region into 14,704 blocks. We stopped the WCVT
balancing process when every zone was connected, all CLFs fell within the range \([0.9, 0.97]\), and the largest
distance between any zone’s generator and centroid was below 0.05 miles. The process terminated after 50
iterations in 4 minutes, 56 seconds of runtime. Table \(C1\) in Appendix \(C\) records the evolution of the WCVT
process in terms of the CLF across zones.

The final average CLF, 0.9382, nearly coincides with the ratio between \(V_{CA}\) and the actual fleet size used,
\(21.65/23 \approx 0.9413\). Figure \(7c\) displays the final partition with zones shaded by CLF and zone centroids
marked. Individual zone areas range from 2.64 square miles to 3.06 square miles (corresponding to zones
15 and 12, respectively). Figure \(7d\) displays the final partition with zones shaded by the expected number of
orders placed within each zone between 8 AM and 3:30 PM. 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 \(A_D(\rho)\). In this case, every vehicle dispatches three times per day.

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. As orders arrive, we solve for the duration of the optimal TSP tour over the current
unserved order set and depot. The vehicle is first dispatched when the sum of the current time, the dispatch
time (given by the TSP duration plus setup and service times), and continuous approximations of future
dispatches sum to \(T\). More formally, if the currently calculated dispatch time is \(\tau\), the vehicle departs at the
latest time \(t_1\) that satisfies \(t_1 + \tau + f_\rho(A, \tau) + f_\rho^{(2)}(A, \tau) = T\), or at the current time if an order arrival leads
to the previous equation having no solution; in this latter case, we do not serve the newest order with this
dispatch. We calculate the second and third departure times in an analogous manner. At each dispatch, the
vehicle takes all accumulated unserved orders, except possibly the immediately arriving order that triggered
the dispatch.

For the vehicle’s 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 order cutoff time for the zone, which may occur before or after
the target cutoff time of 3:30 PM. The cutoff time necessarily varies across zones and from day to day,
reflecting the operational necessities of a system where all orders are planned to be delivered no later than the end-of-day deadline $T$. We simulate a total of 100 service day realizations using this operational policy, and 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 3:30 PM is 4,800. In our simulations, the operational policy serves 5,039.92 orders per day on average with a standard deviation of 36.23 orders, an increase of approximately 5% relative to the expected quantity (which corresponds roughly to the inflation of the vehicle fleet size estimate). 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 clear downward trend in the value of this ratio is evident as the zone CLFs increase.

The planned cutoff time is 3:30 PM, 7.5 hours after the start of the service day. In our simulations, the mean cutoff time was 7.88 hours after the start of the service day (approximately 3:53 PM) with an average standard deviation of 0.31 hours across zones. Each zone cut off orders at or later than 3:30 PM an average of 86.78 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 3:30 PM on average. The trend line implies that a zone with a CLF of 1 would be expected to cut off orders at approximately 3:34 PM on average. This represents a deviation of less than 1% from the target cutoff time of 3:30 PM, suggesting that our continuous approximation approach characterizes this setting with very high accuracy.

The relationship in Figure 9 is nearly linear; the lone outlier at the bottom of the plot corresponds to zone 12. Recall that the dispatch function assumes the vehicle travels to and from the centroid of each zone before and after serving orders via the inclusion of the term $2\rho$; this term generally leads to an overestimation of the actual travel time since the vehicle is unlikely to travel to the centroid of the zone before serving orders in an optimal tour. However, this is not the case when the depot is close to the centroid of a zone, which
Figure 9: Average order cutoff times, central depot

is the situation faced by zone 12. In such instances, the vehicle likely has to travel some positive distance to serve the first order in the tour, but $2\rho \approx 0$ may often underestimate this distance. Hence, we expect the average order cutoff time of zone 12 to be earlier than other zones with similar CLFs.

5.1.2 Sensitivity and managerial insights
The continuous approximation fleet sizing approach facilitates tactical decision-making even without constructing partitions and conducting simulations; we illustrate three 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.

Figure 10: Fleet size sensitivity, central depot
order rate, respectively.

Additionally, our approach may be adapted to determine the size of the service region given a constraint on the number of vehicles available. In the context of this specific region, we may be interested in the value of $V_{CA}$ as the region grows (or shrinks) assuming that the depot remains in the center of the zone and the shape of the zone is preserved. Figure 10c shows how the fleet size guideline varies with the area of the service region under these assumptions.

5.2 Depot located in corner

In our first study, our final partition was nearly symmetric and zone shapes were approximately diamonds due to the simple shape of the service region and central location of the depot. We now consider an irregularly-shaped service region with a total area of 194.5 square miles. The depot is located in the southwest corner of the region (see Figure 12a).

We again use rectilinear distances, a vehicle speed of 25 miles per hour, and a setup time per dispatch of five minutes, while we increase the per-order service time to 2.5 minutes. The service day extends from 8 AM to 8 PM with an earlier target order cutoff time of 2 PM. Each vehicle is again allowed at most $D_{\text{max}} = 3$ dispatches per day. Orders arrive at a rate of two per hour per square mile. To ensure that the order arrival rate is $\lambda = 1$, we re-scale all parameters to be measured in increments of 30 minutes.

Figure 11: Optimal area functions, corner depot

Figure 11 displays the resulting dispatch time function, re-scaled time parameters, and optimal area functions for up to four dispatches per vehicle. The continuous approximation fleet size guideline is $V_{CA} = 17.13$ vehicles for $D_{\text{max}} = 3$. For comparison, the fleet size guidelines are 25.64, 17.75, and 17.12 for $D_{\text{max}} = 1, 2, \text{and} 4$, respectively. We partition the service region into 1.05$V_{CA} = 17.99 \approx 18$ single-vehicle zones. The “many-vehicle” policy with no partitioning described by [44] instead requires 31 vehicles.
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 12a, are used as the initial WCVT generator locations. To construct the partition, the service region was discretized into 19,625 blocks. 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.
As in the previous experiment, 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, $17.13/18 \approx 0.9516$. Individual zone areas range from 8.18 square miles to 14 square miles (corresponding to zones 14 and 1, respectively). Figure 12b displays the final partition with zones shaded by the expected number of orders placed between 8 AM and 2 PM. 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 dispatch three times each, while the remaining vehicles each dispatch twice.

5.2.1 Simulation results

We again simulate 100 service day realizations with the same operational policy. The expected number of daily orders placed within the service region between 8 AM and 2 PM is 2,334. In our simulations, the operational policy serves 2,468.76 orders per day with a standard deviation of 21.73 orders, an improvement of approximately 5.77% relative to the expected quantity, again similar to the fleet size inflation factor. For each zone, Figure 13 displays the ratio between the daily average quantity served and the expected quantity served versus the zone’s CLF. As in the previous study, the value of this ratio is greater than one for each zone. While the trend is again downward as CLF increases, the trend is noisier than in the previous study due to the variance in zone shapes.

Figure 13: Ratios between average quantity served and expected quantity served, corner depot

The planned cutoff time is 2 PM, 6 hours after the start of the service day. In our simulations, the mean cutoff time was 6.37 hours after the start of the service day (approximately 2:22 PM) with an average standard deviation of 0.32 hours 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 14 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 PM on average.
The trendline implies that a zone with a CLF of 1 would be expected to cut off orders at approximately 2:17 PM on average, albeit with less certainty than in the previous study due to the variance in zone shapes. This represents a deviation of less than 5% from the target cutoff time of 2 PM. Additional data and fleet size sensitivity plots are included in Appendix C.

Figure 14: Average order cutoff times, corner depot

5.2.2 *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., 15, 24, 51, 54]. We next briefly consider the question of benchmarking our operational policy against potential improved policies.

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 small-
Table 1: A posteriori upper bounds

<table>
<thead>
<tr>
<th>Zone</th>
<th>D</th>
<th>Average quantity served</th>
<th>Operational-UB gap</th>
<th>Best solution-UB gap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Oper. policy</td>
<td>Best solution</td>
<td>Upper bound</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>175.97</td>
<td>176.03</td>
<td>201.33</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>143.89</td>
<td>143.96</td>
<td>157.53</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>136.17</td>
<td>136.45</td>
<td>138.73</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>105.85</td>
<td>106.80</td>
<td>107.85</td>
</tr>
</tbody>
</table>

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.

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 are within 2% of our operational benchmark. For the three-dispatch zones, the average bounds are still quite close to the operational benchmark, with gaps of under 15% and 10% for zones 1 and 3, respectively. These larger gaps are likely due to the 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 [24, 44].

6. Conclusions

In this paper, we study the tactical problem of partitioning a single-depot same-day delivery 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

-est three-dispatch zones, respectively), 11, and 14 (the largest and smallest two-dispatch zones, respectively).
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 \cite{22, 33, 34} to the modern context of same-day delivery.

We provide two examples of the fleet sizing and partitioning scheme on service regions with different shapes and depot locations. We compute fleet sizes and show that significantly fewer vehicles are required to use our partitioning approach when compared to the number of vehicles required to implement the policy proposed by \cite{44}, which does not include \textit{a priori} partitioning. 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 and efficiently assessing multiple potential solutions. Overall, our approach creates an actionable plan for producing well-designed vehicle-routing zones which satisfy system requirements.

Our work presents multiple avenues for further exploration. An interesting 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 or temporally inhomogeneous order arrivals, and capacitated vehicles. Our optimal tactical dispatching policies rely on the concavity and monotonicity of BHH-style routing 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 \cite{44} 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 system or further reduce the number of vehicles required.

\textbf{Acknowledgements}

Dipayan Banerjee’s work was supported via a U.S. National Science Foundation Graduate Research Fellowship (DGE-1650044).
References


[38] Pillac V, Gendreau M, Guéret C, and Medaglia A (2013) A review of dynamic vehicle routing prob-


[40] Reyes D, Erera A, and Savelsbergh M (2018) Complexity of routing problems with release dates and


doc/scipy/reference/optimize.html

www.optimization-online.org/DB_HTML/2019/06/7243.html In review.

[45] 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


[47] Ulmer M, Goodson J, Mattfeld D, and Hennig M (2019) Offline–online approximate dynamic pro-
202.


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


Appendix A  Omitted Proofs

A.1 Proof of Lemma 1

Consider an area $A > 0$ and an associated feasible dispatching policy $P = ((t_1, \tau_1), \ldots, (t_D, \tau_D))$ 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 $P' = ((t'_1, \tau'_1), \ldots, (t'_D, \tau'_D)) = ((t_1, \tau_1), \ldots, (t_D, \tau_D))$. Suppose instead that $d > 1$. Let $\varepsilon > 0$ equal the waiting time after dispatch $d$. Define $\tilde{t}_d = t_d + \frac{\varepsilon}{2}$, then define $\tilde{t}_{d-1} = t_{d-1} + \frac{\varepsilon}{3}$, and so on until $\tilde{t}_1 = t_1 + \frac{\varepsilon}{2\rho}$. Set $P' = ((t'_1, \tau'_1), \ldots, (t'_D, \tau'_D)) = ((\tilde{t}_1, \tau_1), \ldots, (\tilde{t}_d, \tau_d), (t_{d+1}, \tau_{d+1}), \ldots, (t_D, \tau_D))$. 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 $f_\rho$ 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 $t'_d + \delta$, $\tau'_d$ by $\tau'_d + \delta$, and $\tau'_{d+1}$ 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 $\hat{A}_\rho(\rho) = A_D(\rho) = A > 0$. 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

$$P_0 = \arg\min_{P \in \mathcal{P}} \{k_P\}.$$
Let $P_0 = \left((t_1, \tau_1), \ldots, (t_D, \tau_D)\right)$ 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, $\sum_{d=1}^d \tau_d = t_d$ for all $d < k$, and $\sum_{i=1}^k \tau_i < t_k$. Define $V = \sum_{i=1}^k \tau_i$, the total size 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 $f_\rho$ is continuous, strictly concave, and increasing in $\tau$ 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 $f_\rho(A, \tau_k - \varepsilon) + f_\rho(A, \tau_k + \varepsilon) < f_\rho(A, \tau_{k-1}) + f_\rho(A, \tau_k)$. Define the policy

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

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 $f_p(A, \tau_k) \geq \tau_k$. Because $t_{k-1} + \tau_k = V$, it follows that $t_{k-1} + f_p(A, \tau_k) \geq V$. As $t_{k-1} + f_p(A, \tau_{k-1}) + f_p(A, \tau_k) = W$, trivially $t_{k-1} + f_p(A, \tau_k) + f_p(A, \tau_{k-1}) = W$. Therefore, the policy

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

is feasible for $A$. Observe that $P_2$ was constructed by “swapping” the dispatch accumulation times $\tau_{k-1}$ and $\tau_k$ (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 definition of $k$ and $P_0$.

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

![Figure A3](image)

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

![Figure A4](image)

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

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

In other words, $h(\tau)$ represents the total accumulation time of the first $k-1$ dispatches if $\tau$ 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 $\nu$ be such that $f_p(A, \tau_k + \nu) = W - V$. Since $\tau_1 > 0$, we have that $\mu > 0$, and since $f_p(A, \tau_k) < W - V$, we have that $\nu > 0$. Define $\delta = \min\{\mu, \nu\}$. Because $f_p$ is strictly increasing and
concave in $\tau$ 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 $\epsilon_1, \ldots, \epsilon_{k-1} > 0$ such that

(i) $h(\tau_1 - \epsilon_1) = t_{k-1} - \delta$, and

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

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

\[ f(A, \tau_{k-1} - \epsilon_{k-1}) + f(A, \tau_k + \delta) < \delta + f(A, \tau_{k-1}) + f(A, \tau_k). \quad \text{(A1)} \]

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 $\delta$ lower than in $P_0$, and the accumulation time of the $k$-th dispatch is $\delta$ 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 $W - t_{k-1} = f_{\rho}(A, \tau_k + \delta) < f_{\rho}(A, \tau_{k-1}) + f_{\rho}(A, \tau_k)$.

Thus, the policy

\[ P_3 = \left( (t_1 - \epsilon_1, \tau_1 - \epsilon_1), (t_2 - \epsilon_1 - \epsilon_2, \tau_2 - \epsilon_2), \ldots, \right. \]

\[ \left. (t_{k-1} - \delta, \tau_{k-1} - \epsilon_{k-1}), (W - f_{\rho}(A, \tau_k + \delta), \tau_k + \delta), (t_{k+1}, \tau_{k+1}), \ldots, (t_D, \tau_D) \right) \]

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.

**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
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 \( A < A_{D-m}(\rho) \leq \hat{A}_D(\rho) \), 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 \( A_D(\rho) = \hat{A}_D(\rho) \), we also have \( A_{D-m}(\rho) = \hat{A}_D(\rho) \). However, in the previous part of the proof, we showed that \( A_{D-m}(\rho) < \hat{A}_D(\rho) \), a contradiction.

Thus, we have shown that \( \mathcal{P} = \emptyset \) as desired. \( \square \)

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

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

Simply rearranging this equation gives

\[
\tau = \left( \frac{t - 2\rho}{\gamma A} \right)^2,
\]

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

\[
\beta A x^2 + \gamma A x + (2\rho - t) = 0.
\]

Solving this quadratic equation for \( x \) gives

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

which implies

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

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

A.4 Proof of Theorem 6

We prove the general claim by induction with \( D = 2 \) as the base case, proved earlier. Consider an area maximization problem with \( \alpha = 0 \), \( \rho \in [0, \frac{T-N}{2}] \) and some \( D \geq 3 \) such that \( \frac{N}{\tau} \geq \frac{D-1}{D} \). It follows that \( 2\rho <
Assume, for the purposes of induction, that $A_{D-1}(\rho) > A_{D-2} > \cdots > A_1(\rho)$ since $\frac{D-1}{D-2} < \frac{D}{D-1} \leq \frac{N}{T}$.

The maximum $(D-1)$-dispatch area $A_{D-1}(\rho)$ is associated with some optimal $(D-1)$-dispatch policy $\{(t_1, \tau_1), \ldots, (t_{D-1}, \tau_{D-1})\}$. Because $A_{D-1}(\rho) = \hat{A}_{D-1}(\rho)$, 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 $D = 3$: 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 $D = 3$ dispatches

It follows that $\{(0,0), (t_1, \tau_1), \ldots, (t_{D-1}, \tau_{D-1})\}$ is a feasible $D$-dispatch policy for the area $A_{D-1}(\rho)$ that includes waiting after the first dispatch. Therefore, $A_D(\rho) > A_{D-1}(\rho) > A_{D-2} > \cdots > A_1(\rho)$. 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 $A_D(\rho^*) = 0 < A_1(\rho^*)$. \(\square\)
Appendix B  Partitioning algorithms

B.1 Initial generator selection

Step 0. Randomly initialize \( m \) disk centers \( c_1, \ldots, c_m \) within the region. Initialize \( p = 0, \kappa = \kappa_0, \) and \( \mu = \mu_0. \)

Initialize disk areas to \( A(c_i). \)

Step 1. Iterate \( p \leftarrow p + 1. \) Update the area of each disk as \( \kappa \cdot A(c_i). \) Calculate all pairwise disk overlaps.

If all overlaps are under the overlap tolerance and all disks are fully contained in the service region, terminate the procedure.

Step 2. Calculate the total force vector \( \vec{F}(i) \) on each disk \( C_i \) as the sum of boundary forces, overlap forces, and random perturbations.

Step 3. Update the location of each disk center as \( c_i \leftarrow c_i + \mu \cdot \vec{F}(i). \)

Step 4. Set \( \mu \leftarrow \Delta \mu \cdot \mu. \) If \( p \mod p_{\text{max}} = 0, \) set \( \kappa \leftarrow \Delta \kappa \cdot \kappa, \) set \( \mu \leftarrow \mu_0, \) and set \( p = 0. \) Return to Step 1.

B.2 WCVT balancing

Step 0. For each \( i \in [m], \) initialize the generator location \( z_i \leftarrow c_i \) and weight \( w_i = 1. \)

Step 1. Set \( Z_1, \ldots, Z_m = \emptyset. \)

Step 2. Calculate the target areas for each zone given the current generator locations: for each \( i \in [m], \) set \( a_i \leftarrow A(z_i) \cdot \text{area}(R) \sum_{j=1}^{m} A(z_j). \)

These values represent “ideal” areas corresponding to every zone having the same CLF.

Step 3. Create the weighted Voronoi diagram. First, for each block \( k \in [\ell], \) find its closest generator point, adjusted for weight and maximum area:

\[
\hat{i}^* = \arg\min_{i \in [m]} \left\{ \frac{w_i}{a_i} \| z_i - b_k \| \right\}.
\]

Then, add the block to the corresponding zone: \( Z_{i^*} \leftarrow Z_{i^*} \cup B_k. \)

Step 4. Update the generator locations: for each \( i \in [m], \) set \( z_i \leftarrow (1 - \eta)z_i + \eta \cdot \text{centroid}(Z_i). \)
**Step 5.** Update the weight vector. For each $i \in [m]$, set

$$w_i \leftarrow (1 - \eta)w_i + \eta \cdot \frac{\text{area}(Z_i)}{a_i}.$$ 

**Step 6.** (optional) For each $i \in [m]$, if $\frac{\text{area}(Z_i)}{A(z_i)} > \Theta^+$, set $w_i \leftarrow w_i + \theta^+$. Alternatively, if $\frac{\text{area}(Z_i)}{A(z_i)} < \Theta^-$, set $w_i \leftarrow w_i - \theta^-$. 

**Step 7.** If the stopping conditions are met, terminate the algorithm. Otherwise, return to Step 1.
Appendix C  Experimental Data and Formulations

C.1 Evolution of WCVT processes

Table C1: Evolution of the WCVT balancing process, central depot

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Centroidal load factor</th>
<th>Minimum</th>
<th>Average</th>
<th>Maximum</th>
<th>Std. dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>0.8246</td>
<td>0.9380</td>
<td>1.0759</td>
<td>0.0842</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>0.9001</td>
<td>0.9383</td>
<td>0.9808</td>
<td>0.0202</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>0.8997</td>
<td>0.9382</td>
<td>0.9817</td>
<td>0.0245</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>0.9011</td>
<td>0.9383</td>
<td>0.9688</td>
<td>0.0214</td>
</tr>
<tr>
<td>40</td>
<td></td>
<td>0.8958</td>
<td>0.9382</td>
<td>0.9823</td>
<td>0.0195</td>
</tr>
<tr>
<td>50</td>
<td></td>
<td>0.9100</td>
<td>0.9382</td>
<td>0.9693</td>
<td>0.0198</td>
</tr>
</tbody>
</table>

Table C2: Evolution of the WCVT balancing process, corner depot

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Centroidal load factor</th>
<th>Minimum</th>
<th>Average</th>
<th>Maximum</th>
<th>Std. dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>0.6177</td>
<td>0.9483</td>
<td>1.4663</td>
<td>0.2270</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>0.8801</td>
<td>0.9483</td>
<td>1.0698</td>
<td>0.0476</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>0.9093</td>
<td>0.9483</td>
<td>0.9772</td>
<td>0.0230</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>0.9086</td>
<td>0.9483</td>
<td>0.9819</td>
<td>0.0234</td>
</tr>
<tr>
<td>40</td>
<td></td>
<td>0.8961</td>
<td>0.9483</td>
<td>0.9894</td>
<td>0.0245</td>
</tr>
<tr>
<td>50</td>
<td></td>
<td>0.9101</td>
<td>0.9483</td>
<td>0.9856</td>
<td>0.0226</td>
</tr>
</tbody>
</table>
C.2 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 3:30 PM in the first study, and 8 AM to 2 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 C3: Computational study results, central depot**

<table>
<thead>
<tr>
<th>Zone</th>
<th>CLF</th>
<th>Area (sq. mi.)</th>
<th>Expected</th>
<th>Avg.</th>
<th>Std. dev.</th>
<th>Cutoff (hrs. after 8 AM)</th>
<th># cutoffs after 3:30 PM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Avg.</td>
<td>Std. dev.</td>
</tr>
<tr>
<td>1</td>
<td>0.9693</td>
<td>2.81</td>
<td>210.61</td>
<td>216.41</td>
<td>6.86</td>
<td>7.75</td>
<td>0.28</td>
</tr>
<tr>
<td>2</td>
<td>0.9648</td>
<td>2.79</td>
<td>209.46</td>
<td>215.94</td>
<td>7.64</td>
<td>7.80</td>
<td>0.28</td>
</tr>
<tr>
<td>3</td>
<td>0.9683</td>
<td>2.80</td>
<td>210.25</td>
<td>216.92</td>
<td>7.69</td>
<td>7.75</td>
<td>0.27</td>
</tr>
<tr>
<td>4</td>
<td>0.9471</td>
<td>2.74</td>
<td>205.64</td>
<td>214.34</td>
<td>8.28</td>
<td>7.85</td>
<td>0.30</td>
</tr>
<tr>
<td>5</td>
<td>0.9139</td>
<td>2.65</td>
<td>199.03</td>
<td>211.40</td>
<td>7.07</td>
<td>8.02</td>
<td>0.29</td>
</tr>
<tr>
<td>6</td>
<td>0.9222</td>
<td>2.81</td>
<td>211.02</td>
<td>223.14</td>
<td>7.71</td>
<td>7.98</td>
<td>0.32</td>
</tr>
<tr>
<td>7</td>
<td>0.9290</td>
<td>2.83</td>
<td>212.35</td>
<td>226.71</td>
<td>7.48</td>
<td>7.91</td>
<td>0.31</td>
</tr>
<tr>
<td>8</td>
<td>0.9236</td>
<td>2.82</td>
<td>211.35</td>
<td>223.76</td>
<td>7.38</td>
<td>7.97</td>
<td>0.32</td>
</tr>
<tr>
<td>9</td>
<td>0.9250</td>
<td>2.68</td>
<td>201.34</td>
<td>211.67</td>
<td>8.03</td>
<td>8.02</td>
<td>0.29</td>
</tr>
<tr>
<td>10</td>
<td>0.9226</td>
<td>2.68</td>
<td>200.81</td>
<td>213.92</td>
<td>7.33</td>
<td>7.96</td>
<td>0.30</td>
</tr>
<tr>
<td>11</td>
<td>0.9137</td>
<td>2.80</td>
<td>209.69</td>
<td>225.34</td>
<td>7.21</td>
<td>8.03</td>
<td>0.27</td>
</tr>
<tr>
<td>12</td>
<td>0.9478</td>
<td>3.06</td>
<td>229.36</td>
<td>236.03</td>
<td>6.24</td>
<td>7.67</td>
<td>0.35</td>
</tr>
<tr>
<td>13</td>
<td>0.9618</td>
<td>2.95</td>
<td>221.02</td>
<td>230.95</td>
<td>7.33</td>
<td>7.77</td>
<td>0.27</td>
</tr>
<tr>
<td>14</td>
<td>0.9106</td>
<td>2.64</td>
<td>198.29</td>
<td>211.89</td>
<td>8.09</td>
<td>8.04</td>
<td>0.30</td>
</tr>
<tr>
<td>15</td>
<td>0.9100</td>
<td>2.64</td>
<td>198.03</td>
<td>212.23</td>
<td>7.69</td>
<td>7.98</td>
<td>0.36</td>
</tr>
<tr>
<td>16</td>
<td>0.9209</td>
<td>2.81</td>
<td>210.69</td>
<td>224.37</td>
<td>7.52</td>
<td>7.95</td>
<td>0.32</td>
</tr>
<tr>
<td>17</td>
<td>0.9499</td>
<td>2.89</td>
<td>217.02</td>
<td>227.12</td>
<td>8.01</td>
<td>7.83</td>
<td>0.32</td>
</tr>
<tr>
<td>18</td>
<td>0.9424</td>
<td>2.87</td>
<td>215.35</td>
<td>226.99</td>
<td>7.47</td>
<td>7.84</td>
<td>0.29</td>
</tr>
<tr>
<td>19</td>
<td>0.9204</td>
<td>2.67</td>
<td>200.37</td>
<td>211.94</td>
<td>7.94</td>
<td>8.00</td>
<td>0.35</td>
</tr>
<tr>
<td>20</td>
<td>0.9653</td>
<td>2.79</td>
<td>209.61</td>
<td>216.32</td>
<td>7.81</td>
<td>7.71</td>
<td>0.34</td>
</tr>
<tr>
<td>21</td>
<td>0.9549</td>
<td>2.76</td>
<td>207.25</td>
<td>214.43</td>
<td>7.83</td>
<td>7.79</td>
<td>0.36</td>
</tr>
<tr>
<td>22</td>
<td>0.9423</td>
<td>2.73</td>
<td>204.46</td>
<td>213.72</td>
<td>7.61</td>
<td>7.88</td>
<td>0.27</td>
</tr>
<tr>
<td>23</td>
<td>0.9526</td>
<td>2.76</td>
<td>206.99</td>
<td>214.38</td>
<td>8.51</td>
<td>7.79</td>
<td>0.33</td>
</tr>
<tr>
<td>Avg.</td>
<td>0.9382</td>
<td>2.78</td>
<td>208.70</td>
<td>219.13</td>
<td>7.56</td>
<td>7.88</td>
<td>0.31</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>64</td>
<td>4800</td>
<td>5039.92</td>
<td>36.23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>CLF</td>
<td>Area (sq. mi.)</td>
<td>Orders served</td>
<td>Cutoff (hrs. after 8 AM)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
<td>----------------</td>
<td>---------------</td>
<td>-------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Expected</td>
<td>Avg.</td>
<td>Std. dev.</td>
<td>Avg.</td>
<td>Std. dev.</td>
</tr>
<tr>
<td>1</td>
<td>0.9850</td>
<td>14.00</td>
<td>168.00</td>
<td>175.97</td>
<td>4.11</td>
<td>6.19</td>
<td>0.30</td>
</tr>
<tr>
<td>2</td>
<td>0.9700</td>
<td>12.46</td>
<td>149.52</td>
<td>159.57</td>
<td>5.12</td>
<td>6.41</td>
<td>0.27</td>
</tr>
<tr>
<td>3</td>
<td>0.9702</td>
<td>11.42</td>
<td>137.04</td>
<td>143.89</td>
<td>4.81</td>
<td>6.36</td>
<td>0.32</td>
</tr>
<tr>
<td>4</td>
<td>0.9246</td>
<td>9.96</td>
<td>119.50</td>
<td>126.25</td>
<td>5.29</td>
<td>6.38</td>
<td>0.31</td>
</tr>
<tr>
<td>5</td>
<td>0.9856</td>
<td>13.41</td>
<td>160.92</td>
<td>169.72</td>
<td>5.27</td>
<td>6.30</td>
<td>0.31</td>
</tr>
<tr>
<td>6</td>
<td>0.9651</td>
<td>12.30</td>
<td>147.60</td>
<td>156.80</td>
<td>6.33</td>
<td>6.48</td>
<td>0.33</td>
</tr>
<tr>
<td>7</td>
<td>0.9548</td>
<td>10.83</td>
<td>129.96</td>
<td>136.17</td>
<td>5.81</td>
<td>6.24</td>
<td>0.30</td>
</tr>
<tr>
<td>8</td>
<td>0.9360</td>
<td>9.81</td>
<td>117.72</td>
<td>123.96</td>
<td>6.63</td>
<td>6.43</td>
<td>0.30</td>
</tr>
<tr>
<td>9</td>
<td>0.9393</td>
<td>9.04</td>
<td>108.50</td>
<td>114.89</td>
<td>5.17</td>
<td>6.38</td>
<td>0.30</td>
</tr>
<tr>
<td>10</td>
<td>0.9572</td>
<td>11.74</td>
<td>140.88</td>
<td>151.68</td>
<td>5.22</td>
<td>6.46</td>
<td>0.31</td>
</tr>
<tr>
<td>11</td>
<td>0.9296</td>
<td>10.66</td>
<td>127.92</td>
<td>134.83</td>
<td>6.25</td>
<td>6.36</td>
<td>0.32</td>
</tr>
<tr>
<td>12</td>
<td>0.9461</td>
<td>9.50</td>
<td>114.05</td>
<td>119.60</td>
<td>5.54</td>
<td>6.32</td>
<td>0.38</td>
</tr>
<tr>
<td>13</td>
<td>0.9294</td>
<td>8.80</td>
<td>105.55</td>
<td>112.69</td>
<td>5.75</td>
<td>6.43</td>
<td>0.31</td>
</tr>
<tr>
<td>14</td>
<td>0.9118</td>
<td>8.18</td>
<td>98.13</td>
<td>105.85</td>
<td>6.49</td>
<td>6.49</td>
<td>0.37</td>
</tr>
<tr>
<td>15</td>
<td>0.9622</td>
<td>11.94</td>
<td>143.28</td>
<td>151.81</td>
<td>5.33</td>
<td>6.46</td>
<td>0.31</td>
</tr>
<tr>
<td>16</td>
<td>0.9284</td>
<td>10.48</td>
<td>125.78</td>
<td>133.27</td>
<td>6.51</td>
<td>6.33</td>
<td>0.36</td>
</tr>
<tr>
<td>17</td>
<td>0.9641</td>
<td>10.55</td>
<td>126.60</td>
<td>130.97</td>
<td>5.58</td>
<td>6.28</td>
<td>0.29</td>
</tr>
<tr>
<td>18</td>
<td>0.9101</td>
<td>9.42</td>
<td>113.05</td>
<td>120.84</td>
<td>5.79</td>
<td>6.45</td>
<td>0.33</td>
</tr>
<tr>
<td>Avg.</td>
<td>0.9483</td>
<td>10.81</td>
<td>129.67</td>
<td>137.15</td>
<td>5.61</td>
<td>6.37</td>
<td>0.32</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>194.5</td>
<td>2334</td>
<td>2468.76</td>
<td>21.73</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table C4: Computational study results, corner depot
C.3 Sensitivity plots

Figure C1: Fleet size sensitivity, corner depot

![Graphs showing sensitivity plots](image)

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

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

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 $E = \{ \{i, j\} \mid 0 \leq i < j \leq L \}$
- 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 $\alpha$, and per-order service time $\beta$

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:

\[
\begin{align*}
\text{max} & \quad \sum_{d=1}^{D} \sum_{i=1}^{L} y_i^d \\
\text{s.t.} & \quad \sum_{e \in \delta(0)} x_e^d = 2 & \forall d \in [D], \quad (C1) \\
& \quad \sum_{e \in \delta(i)} x_e^d = 2y_i^d & \forall i \in [L], d \in [D], \quad (C2) \\
& \quad \sum_{d=1}^{D} y_i^d \leq 1, & \quad (C3) \\
& \quad \sum_{d=1}^{D} y_i^{d+1} \leq \sum_{d=1}^{D} y_i^d & \forall i \in [L], \quad (C4) \\
& \quad t_d + \sum_{e \in E} (c_e + \beta)x_e^d - \beta + \alpha \leq t_{d+1} & \forall d \in [D-1], \quad (C5) \\
& \quad t_D + \sum_{e \in E} (c_e + \beta)x_e^D - \beta + \alpha \leq T, & \quad (C6) \\
& \quad t_d \geq r_i y_i^d & \forall i \in [L], d \in [D], \quad (C7) \\
& \quad \sum_{e \subseteq S} x_e^d \leq |S| - 1 & \forall \emptyset \neq S \subseteq [L], d \in [D], \quad (C8) \\
& \quad x_e^d \in \{0, 1\} & \forall e \in E, d \in [D], \quad (C9) \\
& \quad y_i^d \in \{0, 1\} & \forall i \in [L], d \in [D]. \quad (C10)
\end{align*}
\]

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.