Tactical Design of Same-Day Delivery Systems

Alexander M. Stroh       Alan L. Erera       Alejandro Toriello

H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332
alexmstroh@gatech.edu, {aerera, atoriello}@isye.gatech.edu

June 3, 2020

Abstract

We study tactical models for the design of same-day delivery (SDD) systems. Same-day fulfillment in e-commerce has seen substantial growth in recent years, and the underlying management of such services is complex. While the literature includes operational models to study SDD, they tend to be detailed, complex, and computationally difficult to solve, and thus may not provide any insight into tactical SDD design variables and their impact on the average performance of the system. We propose a simplified vehicle dispatching model that captures the “average” behavior of an SDD system from a single stocking location by utilizing continuous approximation techniques. We analyze the structure of optimal vehicle dispatching policies given our model for two important instance families, the single-vehicle case and the case in which the delivery fleet is large, and develop techniques to find these policies that require only simple computations. We also leverage these results to analyze the case of a finite fleet, proposing a heuristic policy with a worst-case approximation guarantee. We then demonstrate with several example problem settings how this model and these policies can help answer various tactical design questions, including how to select a fleet size, determine an order cutoff time, and combine SDD and overnight order delivery operations. We validate model predictions empirically against a detailed operational model in a computational case study using geographic and census data for the northeastern metro Atlanta region, and we demonstrate that our model predicts the average number of orders served and dispatch time to within 1%.

1 Introduction

Total annual retail sales in the United States grew by an estimated 2.76% from 2015 to 2016, in part via electronic shopping, which increased by 12.40% [6]. Within the growing space of e-commerce, same-day delivery (SDD) services are commonly offered by large retailers and logistics providers. A survey of over 500 North American retailers found that 51% claimed to provide SDD fulfillment options in 2017, up from the 16% reported in 2016 [11]. Amazon, one of the current SDD industry leaders, began offering an SDD fulfillment option in October 2009 across 7 major U.S. cities [16]. By April 2016, Amazon offered the service to over 1,000 cities and towns, a figure which rose to over 8,000 by November 2018 [17] [18]. These statistics highlight recent demand increases in the e-commerce space, as well as the adoption of SDD systems by many retailers.

Managing an SDD system is potentially problematic for retailers with already thin profit margins. SDD systems inherit and exacerbate many of the issues faced by more traditional two-day or next-day last mile logistics systems, including tight deadlines, low order volumes, and a high level of order variability and dy-
namism; in general, the uncertainty increases and the time to react decreases \cite{22,23}. Therefore, dispatching and routing orders may be costly and inefficient if not planned carefully.

Like more traditional e-retail delivery services, SDD requires two core logistics processes: order management at the stocking location, including receiving, picking, and packing; and order distribution from the stocking location to customer delivery addresses. We focus here on the second of these processes. Order distribution requires operational decisions, such as when to dispatch a delivery vehicle (timing), and which subset of awaiting customers it will serve (composition). There are clear trade-offs between the timing and composition of SDD dispatches. In some cases, it may be best for a delivery vehicle to wait as long as possible at the stocking location, allowing the accumulation of orders and greater routing efficiency upon dispatch. Alternatively, shorter, more time-inefficient trips could be made in order to reduce the workload within the system, leaving more flexibility to serve orders later in the service day. Such decisions are made numerous times, across a fleet of vehicles, during each service day. Even when the orders to be loaded onto each vehicle for a dispatch are known, deciding the sequence in which to visit delivery locations is a traveling salesman problem (TSP) with possible side constraints; the problem of order assignment and routing in e-commerce has only recently attracted attention from the research community, e.g. \cite{25}. Additionally for SDD, retailers often constrain themselves to serve all SDD demand in a given service day, and thus restrict the latest possible time SDD orders can be placed \cite{12,19}. These order cutoff times can be static or determined dynamically.

Over the past few years, the logistics research community has proposed and studied operational policies for SDD distribution systems using a variety of models and assumptions, e.g. \cite{21,22,23,33,36}. The models considered in the literature to date typically assume a fixed SDD system design, including service area, delivery vehicle fleet size, service time window, etc., and then perform a detailed analysis, optimization and/or simulation of operating policies.

In contrast, the logistics research community has not focused its analysis on the tactical design decisions important for SDD distribution: How large should the SDD delivery vehicle fleet be? How late in the day should SDD service be offered to customers? How large should the service area be? To our knowledge, no papers in the literature address these and other important questions, and our goal is to offer a first attempt. While detailed operational models can in principle offer some insights about such tactical decisions, their granularity implies significant complexity, which in turn renders tactical analysis difficult and less transparent – the models have “too many moving parts”. One goal of this paper is to develop simplified models of operational decisions while maintaining fidelity at the aggregate level; we propose models that do not attempt to capture each order realization and operational decision, but rather to capture the system behavior “on average” so that we may approximate the impact of various design choices on day-to-day operations.

We develop a distribution modeling approach for a single SDD stocking location or dispatch facility, where orders are packed for last-mile delivery and dispatched on delivery vehicles. As in most e-retail settings, we assume orders are customer-specific and cannot be packed or dispatched preemptively, but rather only after they are placed. We also assume a common delivery deadline, e.g. the end of the business day, rather than order-specific deadlines more common in food delivery services \cite{29}. Since SDD systems face tight delivery deadlines and comparatively low order volume, time (not vehicle capacity) tends to be the limiting resource and one of the primary constraints in our models. We initially assume uncapacitated vehicles, and later show that our results extend to the capacitated case with only slight modifications.

To build a simplified SDD dispatch model that still accurately captures system performance, we use a continuous approximation approach in which the expected durations of vehicle routing tours are approximated using a concave, increasing function of the number of orders served. The use of such approximations is well established in logistics \cite{15}, with some canonical results dating back several decades \cite{3,13,27}. When order locations are randomly distributed in the service region according to a continuous distribution, continuous approximations are known to be quite accurate. Such approximations have recently been successfully applied in a last-mile operational context \cite{34}, and can also be calibrated with empirical ob-
servations (see, e.g., [20]). We provide our own computational validation of the approximation model and dispatching policies we develop, and we show that they are remarkably accurate when compared to much more detailed operational models.

Furthermore, although our model is motivated by SDD, our main results rely only on the concavity and monotonicity of the expected dispatch time required to serve a number of orders. Our model could therefore also have applications in other areas where expected processing time is concave and increasing, including batched queueing and warehousing systems, see e.g. [10].

1.1 Contributions

We formulate an SDD dispatching model based on continuous approximations with a single depot and its fleet serving SDD orders in a specified service region. We study the structure of optimal dispatch policies for such models, and make the following specific contributions.

(1) We propose a simple model for SDD dispatching that captures aggregate SDD system behavior by leveraging the elegant structure of continuous approximations. To our knowledge, this is the first such use of this methodology in SDD applications.

(2) We use the dispatching model to analyze two important operational cases for SDD fleets. The first case is when a single vehicle is assigned to a service area and is dispatched multiple times during the operating day, and the second case is when the fleet is large enough that each vehicle is dispatched once per day. We characterize the structure of optimal dispatching policies for these two cases using our model, and show that the optimal policies can be determined using very simple computational techniques, such as finding the roots of equations with single unknowns. Although the case in which multiple vehicles must make multiple dispatches is significantly more difficult to optimize, we propose a heuristic policy for this case with a worst-case performance guarantee.

(3) We use the simple dispatching approximation model and the optimal policies that result to answer various tactical system design questions, including fleet sizing, length of service window, and whether SDD orders should be combined with overnight orders. In all cases, our conclusions rely on simple and transparent analytical properties. The model predictions are validated in a computational study against a detailed operational model using realistic data.

The remainder of the paper is organized as follows. Section 1 concludes with a literature review. In Section 2 we formulate a continuous approximation model of vehicle dispatch operations from a single stocking location and justify our model assumptions. We then describe optimal dispatch policies for specific instances of the proposed model in Section 3. Section 4 provides a managerial analysis of tactical SDD system design using our model and its solutions. We detail a realistic computational experiment using the model and policies in Section 5 and conclude in Section 6. An appendix contains proofs omitted from the main body.

1.2 Literature Review

SDD models can be classified within the rich family of vehicle routing problems (VRPs). The defining features of an SDD model include stochastic order arrivals, order cut-off times and/or a delivery deadline, and perhaps most salient, the overlap in time of dispatching and order arrivals. Examples of model objectives are maximizing expected orders served in a service day, minimizing penalties from undelivered orders, and/or minimizing total routing distance or time given that most or all orders are served. For these reasons, we reference the VRP with probabilistic customer arrivals as studied in [1, 7, 35]. Additionally, dynamic vehicle routing problems such as [4, 26] broadly encompass SDD modeling.
We now survey some operational SDD models from the literature. One such problem is the dynamic dispatch waves problem (DDWP) [22][23]. The DDWP discretizes the dispatch decision epochs, or “waves”, for an operator managing an SDD vehicle fleet. Customer orders arrive according to a known stochastic process and must be served by the end of the service day, or the operator will receive an order-based penalty. Additionally, all delivery vehicles are constrained to return back to the depot before the end of the service day. The objective of the DDWP is to minimize the sum of expected routing cost and penalty cost. In [23] a deterministic variant of the DDWP is solved, where order arrival locations and times are known exactly, over a 1-dimensional service region using the optimal policy structure found in a dynamic programming formulation. In [22] the same authors model the DDWP in 2-dimensions using an integer-programming approach to solve a deterministic variant. Using these deterministic solutions, the authors compute a priori dispatch policies for stochastic variants. These policies are further expanded to dynamic policies in their respective papers. In [23], optimal dispatch policies for an SDD variant (1-dimensional service region, one vehicle) were found to have the property that once the first dispatch occurred, the vehicle never waited at the depot again. Additionally, the durations of successive dispatches are decreasing.

Another SDD model found in literature is the same-day delivery problem for online purchases (SDDP) [36]. The authors provide a general framework for SDD modeling, using a fleet of delivery vehicles of known size, a fixed cut-off time for SDD orders, a known arrival rate and distribution of orders, and a service time and a delivery time window on each order. Like the DDWP, all vehicles operate from a single depot. Unlike the DDWP, the objective of the SDDP is to maximize the expected number of SDD requests that are fulfilled in a service day. A model of the SDDP as a Markov decision process (MDP) is proposed, and dispatch policies are found via a sample-scenario approach with orienteering subproblems. The authors discuss delaying delivery vehicles at the depot for as long as possible without violating delivery time-window constraints or altering vehicle return times. Such properties were shown to exist in optimal dispatching solutions, allowing restriction of the search space.

The DDWP and the SDDP, as well as other works in the literature [30][31][32][33], tend to study the daily operations of SDD logistics, while assuming implicit or explicit knowledge of tactical system design features. It is possible to use these more complicated models to gain tactical-level insight. In [30] the author performs an analysis of the relationship between fleet size and delivery capacity of the system. The authors in [36] observe how the number of fulfilled orders can increase with fleet size by re-solving their operational SDD model multiple times. Such procedures can be useful for determining managerial decisions. However, they require repeatedly solving complex models, often with heuristic methods that do not guarantee optimal solutions; in contrast, our approach will be to exactly optimize a simplified approximation model.

We use continuous approximations for modeling an SDD system. For a recent survey on continuous approximation models in freight logistics, see [15]. The use of such approximations in the field goes back to the BHH theorem [3], a formula for the expected length of a TSP tour as a function of the number of stops visited when locations are drawn from a continuous distribution over the service region. [13][27] then expanded upon this approximation in an analysis of vehicle routing problems with specified dispatch depots for logistics distribution and collection problems, and studied how different zone shapes affect tour lengths and how to select best zone shapes. The work in [20] considers similar approximation ideas for the Held-Karp TSP bound, and [2] calculates empirical constants for TSP length as a function of the number of stops in a tour; see also [37]. Although our study is the first that applies continuous approximations in SDD, there are other applications of these techniques in urban logistics. For example, the work in [14] considers the efficiency of urban commercial vehicles using continuous approximations, and the techniques are used to study drones in last-mile delivery in [9]. Furthermore, a continuous approximation approach is used in [8] to partition a service region for vehicle routing. Operational models for urban last-mile delivery are considered in [34] and continuous approximations are deployed within an approximate dynamic programming framework for optimization. Continuous approximations are frequently applied in facility location; see [24] for a recent example with applications in SDD and the last mile.
2 Model Formulation

We consider an SDD system in which a single depot serves as a stocking location, and its vehicle fleet serves all orders placed from a defined service region. Orders accumulate throughout the service region over the course of the day, and the dispatcher must ensure that all orders are served by the end of the service day at minimum cost. We next formally define our problem by describing the relevant notation and model elements.

**Service Day:** The first time any vehicle can leave the depot is time $t = 0$, and the end of the service day is $t = T$. For convenience, we refer to the service day as having $T$ units of time.

**Customer Orders and Geography:** Demand for SDD in the service region continuously accumulates over time at a constant rate of $\lambda$ orders per time unit. This demand is served by a fleet of $m$ vehicles, each departing from the single depot. For convenience, we assume without loss of generality that $\lambda = 1$ and all other parameters are appropriately scaled.

**Order Cutoff Time:** Customer orders become ready for dispatch starting at time $t = 0$ and continue until time $t = N < T$ in the service day. The service day begins with no orders requiring delivery, and thus the total number of orders that accumulate over the day is $\lambda N = N$. Depending on the context, we may refer to $N$ as the order cutoff time or the number of orders to serve. We also discuss the problem extension when an initial set of orders is ready at the start of the service day.

**Vehicle Restrictions:** All vehicles must return to the depot by time $t = T$. We do not initially constrain the capacity of any vehicle, nor do we restrict any vehicle to carry an integer number of orders. Vehicles may be dispatched more than once during the service day. We later discuss the model extension with capacitated vehicles.

**Routing Time Function:** The time it takes for a vehicle to serve $n \in \mathbb{R}_{\geq 0}$ orders and return to the depot is given by a function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ with the following properties:

1. $f(0) = 0$,
2. $f(n)$ is concave and increasing.

These properties indicate that serving more orders should take more time, but that there will be a gain in marginal time efficiencies when consolidating orders. An example routing time function is of the form $f(n) = a + bn + c\sqrt{n}$ for $n > 0$. This function includes a constant setup time at the depot, a service time per order, and a BHH routing time [3] between orders; [20] showed computationally that these approximations work well in practice even for small order numbers, assuming order locations are independently drawn from a continuous distribution over the service region.

The objective of our model is to choose a set of feasible dispatches that serves all of the orders while minimizing total routing time incurred by all vehicles. We formally define the $d$-th dispatch as a tuple $(t_d, q_d, i_d)$, where $t_d$ indicates the time when vehicle $i_d$ leaves the depot with an order quantity, $q_d$. We assume without loss of generality that dispatches are ordered by time of dispatch, then if necessary by vehicle index. A set of dispatches $\{(t_d, q_d, i_d)\}_{d=1}^{D}$ is feasible for our model if the following conditions are satisfied:

\[
\sum_{d=1}^{D} q_d = N, \quad (1a)
\]
\[
q_d \geq 0 \quad \forall d, \quad (1b)
\]
Our problem is to choose \( D \geq 1 \) dispatch tuples that minimize \( \sum_{d=1}^{D} f(q_d) \), subject to (1a)-(1f). Constraints (1a)-(1b) guarantee that all dispatches serve a non-negative order number, and that these sum to the total number of orders. Constraint (1c) requires all vehicles to return to the depot by the end of the service day. Constraint (1d) guarantees that any vehicle is performing one dispatch at a time. Constraint (1e) ensures the vehicles dispatch only after the service day begins. Finally, constraint (1f) guarantees that orders are served only after they realize.

### 3 Optimal Policies

We initially focus on two important families of instances of our SDD model, the many-vehicle case and the single-vehicle case. In the former, we assume any number of vehicles can be added to the delivery fleet at negligible cost; this situation applies, for example, when we can allocate vehicles for SDD from other resources. In the latter, we focus on the simplest case in which the fleet is constrained, as this case is already of significant interest operationally \([22, 23]\). For the general case with a finite fleet greater than one, we leverage these results to construct a heuristic dispatching policy that combines the policies used at the two extremes.

#### 3.1 Many Vehicles

We first assume the fleet consists of as many vehicles as we like. Consider the following dispatch policy, and the accompanying theorem. The proof is deferred to the appendix.

**Many Vehicle Policy (MVP)** Starting at time \( t = 0 \), dispatch a delivery vehicle at the moment when it can take all of the realized orders waiting at the depot and return at exactly time \( t = T \). Repeat this process until only a single vehicle is needed to deliver all remaining orders and return to the depot before the end of the service day. Dispatch this last vehicle at time \( t = N \).

**Theorem 1.** The MVP is an optimal dispatch policy for the many-vehicle case. Furthermore, if the number of vehicles used by MVP is \( m^* \), the total routing time used by this policy provides a lower bound for the model objective with fleet size \( m < m^* \).

The optimal times given by the policy are easy to solve for in practice, since all that is required is to solve equations of the form, \( t + f(t) = C \), for different values of \( C \). If more than one vehicle is required, then \( t_1 + f(t_1) = T \), and if more than two vehicles are required then \( t_2 + f(t_2 - t_1) = T \), and so on. For problems with realistic parameters, computational results show that the MVP will often require a reasonably sized fleet. Figure 1 depicts an example MVP dispatch plan with four vehicles. Each curved arc corresponds to a vehicle dispatch, while the preceding horizontal straight line represents the time in which that dispatch’s orders accumulate while the vehicle waits. Line styles are alternated for visual clarity.
Figure 1: Visual representation of an MVP requiring four vehicles.

**Example 2.** A retailer provides SDD service for an 8 mile by 8 mile service region, with an average of 75 orders placed over a 10-hour cutoff time. The retailer operates over a 12-hour service day and has an unrestricted fleet size. We scale time to 8 minutes per time unit, and the model parameters are set to $N = 75, T = 90$. Additionally, suppose that the routing time function is $f(n) = 2.15\sqrt{n} + .13n$, roughly equivalent to a routing time approximation (Manhattan distances [20], with vehicles traveling at 25 miles per hour), plus a service time of 1 minute per order. The MVP returns the optimal solution,

$$
t_1 = 64.38, \quad q_1 = 64.38, \quad i_1 = 1,
$$

$$
t_2 = 75, \quad q_2 = 10.62, \quad i_2 = 2,
$$

with 272.06 total minutes of routing time.

### 3.2 One Vehicle

Now assume the fleet consists of a single delivery vehicle. If $N + f(N) \leq T$, this vehicle can (optimally) wait until time $t = N$ to dispatch once with all $N$ orders. More generally, we use the following lemma below in our analysis. Qualitatively, this lemma suggest that given a fixed amount of orders to be served, it would be better to split the orders between two dispatches as unevenly as possible to maximize routing efficiencies. This result follows directly from the concavity of $f$.

**Lemma 3.** The optimal solution of

$$
\min_{a \leq y \leq b} \{ f(y) + f(Y - y) \}
$$

is $y^* = a$ if $a \leq Y - b$, and $y^* = b$ otherwise, for any $0 \leq a \leq b \leq Y$.

Before stating our main results for a single vehicle, we must address a pathology arising from Lemma 3. Consider an instance with $N = 9, T = 11.99, f(n) = \sqrt{n}$, clearly, a single dispatch is infeasible, since the vehicle would return to the depot too late by 0.01 time units. It can be shown that any feasible two-dispatch policy satisfies $0.06 \leq q_1 \leq 8.25$ and $q_2 = 9 - q_1$. By the lemma, the best solution using two dispatches is given by $q_1^* = 0.06$. Furthermore, this policy is in fact optimal for a single vehicle. There are two characteristics of this optimal policy worth noting. First, the first time of dispatch can be adjusted to any time in the range of $q_1^* \leq t_1^* \leq N - f(q_1^*)$, while $t_2^* = N$. Second, the first dispatch size is very small, which is likely to be unreasonable since the square root approximation of routing time tends to be inaccurate for small numbers. This example shows that the model requires additional constraints to produce meaningful answers. We now introduce three additional assumptions that address these concerns.

**Sufficient processing speed:** There exists $q^* < \infty$ such that $f(x) \leq x/\lambda = x$, for all $x \geq q^*$. 
**Minimum dispatch size:** Any feasible solution \( \{ (t_d, q_d, i_d) \}_{d=1}^{D} \) must satisfy \( q_d \geq q_{\text{min}} \) for all \( d < D \). The parameter \( q_{\text{min}} \) is chosen by the dispatcher such that \( q_{\text{min}} \geq q^* \).

**Sufficient gap time:** The parameters \( T, N, q_{\text{min}} \) must satisfy \( T - N \geq f(2q_{\text{min}}) \).

The first assumption ensures that the system can “keep up” with orders; that is, the time required to serve a number of orders is smaller than the time it takes for them to accumulate when the number is large enough. Similarly, a minimum dispatch size is justifiable economically. We bound this minimum size by \( q^* \), so whenever the delivery vehicle is dispatched, it is guaranteed to arrive back at the depot to find a lesser number of unserved orders; this is desirable since all orders must be served by the end of the day, and also prevents solutions like the one described above. The last dispatch is not subject to this constraint, since it must serve all remaining orders, regardless of their number. Our final assumption ensures that there is enough time between the last order arrival and end of service day. These assumptions are sufficient for feasibility; the next lemma formalizes this argument and is proven in the appendix.

**Lemma 4.** An SDD problem instance with a single vehicle and parameters satisfying the minimal dispatch size, sufficient gap time, and sufficient processing speed conditions has a feasible solution.

We next state our main results on optimal dispatch policies for a single vehicle, which are proved in the appendix.

**Theorem 5.** Assuming the minimal dispatch size, sufficient gap time, and sufficient processing speed conditions hold, there exists an optimal dispatch policy for the SDD problem with one delivery vehicle such that

1. each dispatch takes all available unserved orders at the depot at the time of dispatch,
2. after the first dispatch, the vehicle never waits at the depot again, and
3. if the vehicle is dispatched more than once, the last dispatch arrives back at the depot at exactly time \( t = T \).

Theorem[5] implies that an optimal policy can be described by a single parameter, \( t_1^* \). By (1), we have \( q_1^* = \lambda t_1^* = t_1^* \), and then (2) yields \( t_2^* = t_1^* + f(q_1^*) \). Applying this reasoning recursively, \( q_2^* = \lambda (t_2^* - t_1^*) = t_2^* - t_1^* \) and so on, continuing until the last dispatch covers all the remaining orders, leaving at or after time \( t = N \). This structure implies that an optimal dispatch policy can be computed via an optimization model over the single variable, \( \alpha = t_1 \), the time of first dispatch. We can solve for the optimal time of first dispatch, \( \alpha^* \), via an iterative root-finding algorithm searching over all policies that satisfy (1), (2) and (3); see Algorithm[1] below. By properties (2) and (3) we know that the vehicle will be in use for a time \( T - \alpha \), thus an optimal policy is one for which (1)-(3) are satisfied and \( \alpha \) is maximized. This can be interpreted as finding a routing policy that satisfies (1)-(3) while minimizing the number of dispatches used. The algorithm works by fixing a number of dispatches and attempting to find a feasible time of first dispatch satisfying (1)-(3). If there is no such feasible solution, the number of dispatches is increased by one and the process repeats. The function \( h_d(\alpha) \) outputs the maximum number of orders that can be served if the vehicles leaves at time \( \alpha \) using \( d \) dispatches, and \( \alpha_d \) is the earliest time of first dispatch in which \( d \) dispatches can cover all orders.
Algorithm 1 Calculating the optimal time of first dispatch, $\alpha^*$

1: Set $d \leftarrow 1$, $\alpha_1 \leftarrow N$
2: Define $f^\delta(\alpha) := f(f(\cdots f(\alpha)))$, ($\delta$ times)
3: if $N + f(N) \leq T$ then
   4: Set $\alpha^* \leftarrow N$
   5: Set $\text{dispatches}^* \leftarrow d$
else
   7: while $\text{found} = \text{FALSE}$ do
      8: Set $d \leftarrow d + 1$
      9: Define $h_d(\alpha) \leftarrow \alpha + \sum_{\delta=1}^{d-1} f^\delta(\alpha)$
     10: Set $\alpha_d \leftarrow \arg\min_{\alpha \geq 0} (\alpha)$ subject to $h_d(\alpha) = N$
     11: Set $\alpha^* \leftarrow \arg\max_{\alpha \geq 0} (\alpha)$ subject to $h_d(\alpha) + f(N - h_{d-1}(\alpha)) = T$, $\alpha \in [\alpha_d, \alpha_d-1)$
     12: if $\alpha^*$ exists then
        13: Set $\text{dispatches}^* \leftarrow d$
     14: end if
   15: end while
  16: end if
17: Return $\text{dispatches}^*, \alpha^*$

Example 6. Consider the same instance as in Example 2 that is, $N = 75, T = 90, f(n) = 2.15\sqrt{n} + 13n$. Now suppose the fleet has a single delivery vehicle, and $q_{\min} = 12$. This set of model parameters satisfy the minimal dispatch size, sufficient gap time, and sufficient processing speed conditions, so we can use Theorem 5 and our root finding algorithm to compute an optimal dispatch policy:

We first set $d \leftarrow 1$, and $\alpha_1 \leftarrow 75$. As $N + f(N) = 103.37 > 90 = T$, a single dispatch is insufficient and we must consider using two or more dispatches. For two dispatches, we calculate that $\alpha_2 = 52.58$, and thus consider all policies with two dispatches where $\alpha \in [52.58, 75)$. From here we wish to determine if there is an $\alpha$ in this range satisfying $\alpha + f(\alpha) + f(N - \alpha) = T$. Indeed, $\alpha = 54.65$ solves this expression; thus the algorithm terminates. The calculated optimal policy is

\[
\begin{align*}
    t_1 &= 54.65, & q_1 &= 54.65, & i_1 &= 1, \\
    t_2 &= 77.65, & q_2 &= 20.35, & i_2 &= 1,
\end{align*}
\]

with a total routing time of 282.74 minutes; see Figure 2. By comparing this example to the previous one, we can see that by decreasing the fleet from two to one vehicles we would increase the total routing time by less than 4% (272.06 minutes to 282.74 minutes).

![Figure 2: Visual representation of optimal dispatch policy for Example 6.](image)

3.3 General Fleet Size

We now consider the more complex case in which the fleet is finite but greater than one, and thus many vehicles may need to be dispatched more than once. Unfortunately, it is no longer possible to show that
optimal policies satisfy simple structural properties in this case. For example, consider a family of instances where the fleet has two vehicles, but three dispatches are required to serve the orders feasibly. Depending on the parameters $T$, $N$ and $f$, it is possible to construct both instances where it is optimal for the vehicle dispatched first to return before $T$ and to make a second dispatch while the second vehicle is used only once, but also instances where it is instead optimal for the vehicle dispatched second to make the additional dispatch.

Although the optimization of this case is significantly more complex, we can leverage our analysis of the previous cases to construct a heuristic policy, described next.

**Hybrid Policy**  For a fleet with $m$ vehicles, the first $m-1$ are dispatched according to the MVP policy. The final vehicle serves all remaining orders according to the single-vehicle policy computed with Algorithm 1.

The next result shows that this heuristic policy produces solutions within a worst-case factor of optimality for an important class of routing time functions. The proof is in the Appendix.

**Theorem 7.** Suppose $f(n) = bn + c\sqrt{n}$, where $b \geq 0$ and $c \geq 0$. If the hybrid policy dispatches the last vehicle $D_m$ times, the policy cost is within a factor $\frac{m-1+D_m/\sqrt{D_m}}{m-1+D_m}$ of optimal in the worst case.

See Figure 3 below for an example of the hybrid policy compared to the MVP.

![Figure 3](image.png)

Figure 3: In this example, the MVP uses three vehicles. If only two are available, the hybrid policy stipulates that the first vehicle behaves as in the MVP, while the second performs an optimal single-vehicle dispatch policy on the remaining orders, which in this case require three dispatches.

**Example 8.** Consider again the problem instance from Examples 2 and 6. The single-vehicle policy is a special case of the hybrid policy when $m = 1$ (and thus a single vehicle must make all dispatches even if the MVP uses multiple vehicles). Because the single-vehicle policy uses two dispatches in this instance, we conclude from Theorem 7 that its cost is at most a factor $\sqrt{2} \approx 1.41$ larger than the MVP cost. In this case, we know from direct calculation that this cost difference is much smaller, only around 4%.

The guarantee provided by Theorem 7 improves as $m$ grows, since more of the hybrid policy’s dispatches exactly mimic what the MVP does; for example, with $m = 2$ and $D_2 = 2$, the guarantee improves to $(1 + \ldots$
\[2\sqrt{2}/3 \approx 1.28.\]

4 Model Applications

The discussion in Examples \[2\] and \[5\] demonstrate how our model can be applied for tactical design, specifically in fleet sizing. We next discuss other potential uses of the model.

4.1 Serving the Entire Region versus Partitioning

Location analysis and customer assignment are important strategic and tactical questions in logistics, and continuous approximation models have been successfully applied for service region design, e.g. \[8\]. We can similarly ask in an SDD context whether partitioning the service region offers advantages over simply having every vehicle serve the entire region.

Consider a routing time function \( f(n) = a + bn + c\sqrt{n} \). Suppose we partition the service region into \( m \) sub-regions of equal size, so that the demand arrival rate in each is \( 1/m \); each sub-region would then have a routing time function of the form \( h(n) = a + bn + c\sqrt{n/m} \), since the area the vehicle serves is scaled down by a factor of \( 1/m \). At time \( t = N \), if a single vehicle can serve each sub-region with a single dispatch, the total routing time for all vehicles would be

\[
m \times h(N/m) = m \left( a + b(N/m) + c\sqrt{N/m} \right) = am + bN + c\sqrt{N};
\]

the last two terms correspond exactly to the service and routing time a single vehicle would need to serve all \( N \) orders in a single dispatch. Therefore, if the MVP policy uses \( m^* \) vehicles and it is feasible to partition the region into \( m^* \) sub-regions and serve each with a single dispatch, partitioning is preferable. However, the number of required vehicles for a partitioning strategy with a single dispatch per vehicle may differ from \( m^* \) and be either larger or smaller.

Example 9. A retailer provides SDD service for an 8 mile by 8 mile service region, with an average of 75 orders placed over a 10-hour cutoff time. The retailer operates over an 11 hour and 20 minute service day. We scale time to 8 minutes per time unit, and the model parameters are set to \( N = 75, T = 85 \). Additionally, take the routing time function as \( f(n) = 1.88 + 0.25n + 2.15\sqrt{n} \), roughly equivalent to a routing time approximation (Manhattan distances \[20\], with vehicles traveling at 25 miles per hour), plus a service time of 2 minutes per order and a setup time of 15 minutes. The MVP returns the optimal solution

\[
t_1 = 53.87, \quad q_1 = 53.87, \quad i_1 = 1, \\
t_2 = 70.30, \quad q_2 = 16.43, \quad i_2 = 2, \\
t_3 = 75, \quad q_3 = 4.70, \quad i_3 = 3,
\]

with 428.37 total minutes of routing time. In contrast, the minimum number of vehicles needed for a partition strategy as described above is five, with each delivering 15 orders in a total of 374.16 minutes. The manager must then decide whether saving 54.21 minutes per service day is worth an additional two vehicles in the SDD fleet. We can similarly use our one-vehicle policy to develop partitioning strategies with a single vehicle serving each sub-region but performing multiple dispatches.

4.2 Orders at the Start of the Service Day

Thus far, our model assumes no orders are ready for dispatch at the start of the service day. It may be that the SDD system is also required to serve some next-day or overnight orders. In the model, this translates to a number \( N' \geq 0 \) of orders that are ready at the start of the service day.
In the many vehicle case, the approach is similar to the MVP, with one modification. Let \( Q = f^{-1}(T) \), i.e. \( Q \) is the unique number satisfying \( f(Q) = T \) (the inverse exists and \( Q \) is unique because \( f \) is increasing); this number is implicitly a capacity on the number of orders a vehicle can carry during the service day to remain time-feasible. We can now define a generalized MVP, which returns an optimal policy. The generalized MVP optimality proof follows directly from Theorem 1 and Lemma 3.

**Generalized MVP** At time \( t = 0 \), dispatch as many vehicles as possible each carrying \( Q \) orders. The subsequent dispatches are calculated via the MVP.

If the number of orders available at the start of the day is large, the generalized MVP may not capture additional opportunities for routing efficiency stemming from directly optimizing a vehicle routing problem for these orders; however, such opportunities do not relate to the SDD system and would rely on more established routing models.

In the single vehicle case, it is possible that a problem instance previously defined by \( N, T, q_{\text{min}} \) is still feasible for \( N' > 0 \). Define an augmented problem with \( \bar{N} = N' + N, \bar{T} = N' + T, \) and \( \bar{N}' = 0. \) If the solution to this problem has an initial dispatch quantity of \( \bar{q}_1^* \geq N' \), then the optimal quantities for the original instance are identical to those of the augmented one, with dispatch times moved up by \( N' \).

By relaxing the guaranteed feasibility condition from section 3.2, we can solve the one vehicle problem for any instance of \( N' > 0 \). Practically, this involves increasing the gap between the order cut-off time and the end of the service day.

**Generalized Guaranteed Feasibility:** If the parameters \( T, N, N', q_{\text{min}} \) satisfy \( N + N' \leq T - f(2q_{\text{min}}) \), the instance is feasible with a single dispatch vehicle.

Assume the generalized guaranteed feasibility condition holds for parameters \( T, N, q_{\text{min}}, N' \). As before, define and solve an augmented problem with \( \bar{N} = N' + N, \bar{T} = N' + T, \) and \( \bar{N}' = 0. \) In the case that one dispatch is optimal it is necessary that \( \bar{q}_1^* = \bar{N} \geq N' \). In the case of a multiple dispatch optimal solution, the vehicle will only ever be idle at the depot at the start of the service day and will be done serving orders exactly at time \( \bar{T} \). Thus, the total dispatching time is equal to \( \bar{T} - \bar{q}_1^* = T + N' - \bar{q}_1^* \). Because the original problem over \( N, N', \) and \( T \) is feasible, we know that the total routing time for any optimal policy is less than or equal to \( T \) units of time. Additionally, any feasible solution to the original problem can be implemented in the augmented problem. Therefore the optimal solution to the augmented problem must use less than or equal to \( T \) units of routing time. It follows that \( \bar{q}_1^* \geq N' \). Therefore, in all cases it is true that \( \bar{q}_1^* \geq N' \), which implies that the optimal quantities for the original instance are identical to those of the augmented one, with dispatch times moved up by \( N' \).

### 4.3 Capacitated Vehicles

Compared to traditional delivery settings, SDD systems operate in an environment with reduced order volume and much tighter time constraints. Therefore, in many cases the number of orders that can be delivered while satisfying time constraints is relatively small, and thus vehicle capacity is not a binding constraint. Nevertheless, there may be situations in which capacities must also be considered; we next discuss how our results extend to this case. Extending the notation we use in Section 4.2 suppose that each vehicle has capacity to serve at most \( Q > q_{\text{min}} \) orders. Consider first the many-vehicle case, and the following natural extension to the MVP.

**Capacitated MVP** Compute the MVP; if the first (and largest) dispatch serves \( Q \) or fewer orders, implement the policy. Otherwise, dispatch the first vehicle with \( Q \) orders, update \( T \) and \( N \) by subtracting \( Q \), and recompute the MVP on the updated instance. Repeat until the computed MVP is feasible.
Intuitively, the MVP tries to serve as many orders as possible with each successive dispatch while still having the corresponding vehicle return by the end of the service day. The capacitated version of the policy does the same, but must also respect the additional vehicle capacity. As with the Generalized MVP in the previous section, the optimality of this policy follows directly from Theorem 1 and Lemma 3.

Similar reasoning applies when the fleet has one delivery vehicle; note that the proof of Lemma 4 is not affected by adding a capacity, so the instance remains feasible. As in the many vehicle case, Algorithm 1 constructs a solution in which the first dispatch is the largest; when this dispatch is too large for the capacity, the modification replaces it with the maximum possible quantity and iterates on the remaining, smaller instance.

**Capacitated One-Vehicle Policy** Run Algorithm 1 if \(\alpha^* \leq Q\), implement this solution. Otherwise, update \(T\) and \(N\) by subtracting \(Q\), and run Algorithm 1 on the smaller instance. Repeat until \(\alpha^* \leq Q\) in Algorithm 1. If this process occurs \(k\) times before \(\alpha^*\) is feasible, the solution to the instance has \(k\) dispatches of size \(Q\) before the dispatches given by Algorithm 1.

As a simple example, suppose \(N + f(N) \leq T\) but \(Q < N \leq 2Q\), i.e. a single dispatch is time-feasible but capacity-infeasible, and two dispatches suffice. (Although this is not what we expect in SDD, the setting serves for illustration purposes.) In this case, it is optimal for the first dispatch to take \(Q\) orders, and the second dispatch takes the rest; the two dispatches can take place consecutively, returning at time \(T\). More generally, we are able to give the following guarantee.

**Corollary 10.** Suppose \(Q \geq 2q_{\min}\). The capacitated one-vehicle policy produces an optimal solution to the one-vehicle instance with vehicle capacity \(Q\).

The proof of Theorem 5 uses an exchange argument that still applies in the capacitated version with simple modifications. The requirement \(Q \geq 2q_{\min}\) is a technical condition necessary to make the final step of the proof work; however, in practice it is reasonable to expect that the capacity of a delivery vehicle is at least twice its minimum quantity.

### 4.4 Choosing Order Cutoff Time

Consider again the instance in Example 2. In an optimal solution, the second vehicle has almost 53 minutes of slack between its earliest possible arrival back to the depot and the end of the service day \(T\). The manager could consider either reducing \(N\) so the system requires only one vehicle, or increasing \(N\) to serve more orders with this vehicle and increase its utilization.

For this discussion, we fix \(T, f(\cdot), q_{\min}\) and allow \(N\) to vary. We assume that earned revenue from orders served is proportional to \(N\) with constant \(\beta\), and routing costs are proportional to the routing time of the optimal policy, denoted by \(g(N)\). Without loss of generality, we scale \(\beta\) so we can compare revenue against cost. Suppose the SDD manager wishes to choose a cutoff time that maximizes system profit as measured by earned revenue minus routing operational costs. Equivalently, this cutoff time would also minimize the cost of serving orders that occur before the cutoff plus the opportunity cost of not serving orders after the cutoff. The profit maximizing cutoff time is then given by

\[
\max_{0 \leq N < T} \pi(N) = \beta N - g(N). \tag{2}
\]

First we analyze the many vehicle case. Recall that in the MVP, if the solution uses \(m\) vehicles, the first \(m - 1\) return exactly at time \(T\). If the cutoff time is chosen carefully, the last dispatch also returns precisely at this time. The proof of Theorem 13 follows directly from Theorem 1 and the concavity of \(f(\cdot)\), which implies that \(g(N)\) is piecewise concave.
Theorem 11. For the many vehicle case, in the optimal solution \( N^* \) of (2), the MVP policy solution has all dispatches returning exactly at time \( T \).

Let \( N_i \) be the order number at which the MVP uses exactly \( i \) vehicles, with all vehicles returning exactly at time \( T \). Letting, \( N_0 = 0 \), we have the recursion \( N_i = N_{i-1} + \Delta_i \), where \( \Delta_i \) uniquely solves \( \Delta_i + f(\Delta_i) = T - N_{i-1} \) for all \( i \). With a slight abuse of notation, we can denote \( \pi(i) = \pi(N_i) \) as the profit obtained from completely utilizing \( i \) vehicles; this profit is convex in \( i \), and therefore we can calculate the optimal \( N^* \) by iteratively calculating each \( \pi(N_i) \) and stopping when this quantity decreases.

Now we analyze the one vehicle case, where we impose the upper bound \( U \leq T - f(2q_{\min}) \) on \( N \). Define \( i_{\max} \) as the number of dispatches in the optimal dispatch policy when \( N = U \), and suppose \( i_{\max} \leq 2 \). Define \( \bar{N}_1 \) such that for \( N \in (0, \bar{N}_1] \), the optimal dispatch policy determined via Theorem 5 uses exactly one dispatch.

Proposition 12. In the single vehicle case, for any \( U \) such that \( 0 \leq U \leq T - f(2q_{\min}) \) and \( i_{\max} \leq 2 \), the optimal solution of (2) subject to \( N \leq U \) satisfies \( N^* \in \{0, \bar{N}_1, U\} \).

The proof of the proposition can be found in the appendix. The proof for the general case of \( i_{\max} \) would follow from showing that \( g(N) \) is piecewise concave with breakpoints at each \( \bar{N}_i \). We have empirical evidence that this is indeed the case but have not been able to prove it.

We now return to the instance in Examples 2 and 6, and calculate an optimal value for the cutoff time \( N \).

Example 13. Consider the same instance as in the previous examples with \( T = 90, f(n) = 2.15\sqrt{n} + .13n \), and \( q_{\min} = 12 \). Suppose \( \beta = 0.80 \) and profit is given in scaled monetary units.

For the many vehicle case, we calculated the first four values of \( N_i \) to be \( N_0 = 0, N_1 = 64.38, N_2 = 79.62, \) and \( N_3 = 84.57 \). The associated costs are 0, 25.62, 36.00, and 41.43, which result in profits of \( \pi(N_0) = 0, \pi(N_1) = 25.88, \pi(N_2) = 27.70, \) and \( \pi(N_3) = 26.22 \). Thus, the optimal order cutoff time is \( N^* = N_2 \), with an optimal dispatch policy of \( \{(t_1 = 64.38, q_1 = 64.38, i_1 = 1), (t_2 = 79.62, q_2 = 15.24, i_2 = 2)\} \).

For the one vehicle case, we let \( U = T - f(2q_{\min}) = 76.35 \), and it follows that \( i_{\max} = 2 \). Note that, \( \bar{N}_1 = 64.38 \). The associated costs of \( N^* \in \{0, \bar{N}_1, U\} \) are 0, 25.62, and 36.04, which result in profits of \( \pi(0) = 0, \pi(\bar{N}_1) = 25.88, \) and \( \pi(U) = 25.38 \). Thus, the optimal cutoff time is \( N^* = \bar{N}_1 \) with the corresponding policy \( \{(t_1 = 64.38, q_1 = 64.38, i_1 = 1)\} \).

Figures 4 and 5 plot \( \pi(N) \) for this instance in the many and single vehicle cases, respectively.

![Figure 4: Profit function with respect to cut-off time, many vehicle case](image-url)
5 Computational Study

Our SDD planning model uses continuous approximations to preserve simplicity and transparency of analysis, and the previous sections discuss various ways in which the model can be used to inform managerial decisions about an SDD system, including fleet sizing, the choice of service cutoff time, and so forth. When making such decisions, we naturally depend on the accuracy and fidelity of the model when compared to more granular operational models. We now present a computational case study that empirically demonstrates the accuracy of the model and its potential practical use.

Our study considers a hypothetical SDD system where one service area comprises roughly 26 square miles in northeastern metro Atlanta. Specifically, this service region consists of the 22 census tracts north of Interstate 85, south of Interstate 285 and east of Georgia Highway 400, with a population of 92,198 as measured by the U.S. Census Bureau [5]. For the study, we chose five representative addresses within each tract, for a total of 110 potential customer locations, plus a depot location on the northeast border, on Interstate 285. Figure 6 depicts the service region and the 110 representative customer locations.

We assume SDD orders begin at 9 am, and the service day ends at 6 pm; we evaluate two different order cutoff times below in different experiments. Assuming 5% of the working population in the region would like to use the SDD service once every other month within the service day, an order would arrive on average approximately once every six minutes over this time, and we use this rate in the model. However, we do not assume orders are equally likely to appear anywhere in the region: We assign each tract a weight proportional to the product of its median household income and its population (both available from [5]); assuming an order originates in a tract, one of its five representative locations is chosen uniformly at random.

To construct the routing component of our dispatch time function, we sampled customer locations with replacement according to the weighted distribution described above, with sample sizes ranging from 10 to 75 locations and a total of 1,980 samples. We used the Google Maps API [28] to query driving time between every pair of locations, and for each sample calculated the optimal TSP route time for a vehicle route visiting these locations from the depot. From this, we obtained via linear regression a routing time approximation of $24\sqrt{n}$ minutes for $n$ orders, with an R-squared value of 0.94. Furthermore, we include 1.5 minutes of service time per order and a fixed setup time of 10 minutes per dispatch. After re-scaling the instance parameters to be measured in increments of six minutes, this results in an order arrival rate $\lambda = 1$, service
day length $T = 90$ and dispatch time function

$$f(n) = 1.67 + 0.25n + 4\sqrt{n}.$$  

For our operational simulation, we replace the constant arrival rate with a Poisson arrival process that uses the same rate; as in our calibration experiment, the tract location of an arriving order is chosen randomly with weights proportional to the product of median household income and population, and its location within the tract is chosen uniformly at random. We use actual driving times between pairs of locations given by the Google API and determine the total routing time for a dispatch by solving a TSP using a standard integer programming formulation implemented in Gurobi 7.5.2. Our experiments are coded in Python 3.6.3 and run on a Linux computing cluster, which employs HTCondor 8.8.4.

## 5.1 Many Vehicle Policy

First, we assume that this service region is served by two delivery vehicles each performing a single daily dispatch. Following the reasoning from Section 4.4, we choose a cutoff time to fully utilize both vehicles. With these parameters, the MVP prescribes that the first dispatch takes 48.4 orders and its route requires a duration of just under 250 minutes, while the second dispatch takes 18.2 orders with a duration of approximately 140 minutes; the total dispatch time sums to about 389 minutes. The predicted dispatch time for the second vehicle translates to an order cutoff time of 3:40 PM ($N = 66.7$).

We now describe our simulated operational benchmark for the many vehicle policy. As customer orders arrive according to the Poisson process, we solve a TSP on all currently unserved orders. A vehicle is dispatched to deliver the orders when the dispatch setup, service time for accumulated orders, and the duration of the optimal TSP tour added together match the remaining time in the service day, i.e. each vehicle returns exactly at time $T$. After the first dispatch, subsequently arriving orders are assigned to the second vehicle; once the second vehicle is dispatched, we stop accepting orders, which may occur before or after the nominal cutoff time of $N$; this represents a dynamic modification of the cutoff time at the operational level and is in line with how some SDD services operate in practice.
We simulated the operational benchmark 300 times. For each realization, we record the number of orders served and the dispatch time for each vehicle, respectively. Table 1 reports results; for each quantity we include the prediction of our tactical model, and the sample mean and 95% confidence intervals of the operational benchmark. As the table shows, our tactical model predicts the expected number of orders served and the expected total dispatch time to within less than 1%.

<table>
<thead>
<tr>
<th></th>
<th>Tactical</th>
<th>Operational</th>
<th>A Posteriori</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Dispatch Quantity</td>
<td>48.40 units</td>
<td>48.20 units (± 0.51)</td>
<td>43.90 units (± 0.63)</td>
</tr>
<tr>
<td>First Dispatch Time</td>
<td>249.58 min.</td>
<td>249.69 min. (± 1.81)</td>
<td>228.07 min. (± 2.62)</td>
</tr>
<tr>
<td>Second Dispatch Quantity</td>
<td>18.26 units</td>
<td>18.45 units (± 0.35)</td>
<td>22.75 units (± 0.45)</td>
</tr>
<tr>
<td>Second Dispatch Time</td>
<td>139.95 min.</td>
<td>139.16 min. (± 1.47)</td>
<td>144.88 min. (± 1.48)</td>
</tr>
<tr>
<td>Total Quantity</td>
<td>66.66 units</td>
<td>66.65 units (± 0.71)</td>
<td>66.65 units (± 0.71)</td>
</tr>
<tr>
<td>Total Time</td>
<td>389.53 min.</td>
<td>388.85 min. (± 2.85)</td>
<td>372.95 min. (± 3.29)</td>
</tr>
</tbody>
</table>

Table 1: Computational study results, many vehicle policy.

Because our main goal is tactical design and describing average behavior rather than operational management, we do not consider many potential dispatching improvements or modifications that could be used at the operational stage; the logistics literature has several works dedicated to this question, e.g. [22, 33, 36]. Nevertheless, it is also important to assess the quality of our prescribed solution when compared to what an operational decision support tool could accomplish.

Motivated by this question, we compute an a posteriori or “hindsight-optimal” solution for each simulated realization, which provides a lower bound on the dispatch time any operational policy can achieve. For each realization, we assume that the dispatcher knows in advance the exact time and location of each order served by our operational benchmark and then optimizes the two vehicles’ routes with this knowledge. For example, in one of the realizations the operational benchmark could have both vehicles visiting the same neighborhood to deliver two different orders; the a posteriori solution could use its advance knowledge to shift the first order to be served by the second vehicle (with virtually no increase in its routing time), while deleting this order from the first vehicle’s dispatch would reduce its routing time. This example also illustrates that despite having advance knowledge of order times and locations, the a posteriori solution must still satisfy operational constraints; in particular, an order can only be served by a dispatch that departs the depot after the order ready time, and the vehicles must return to the depot by the end of the service day. In the appendix, we include the formulation we use to compute these solutions.

In our experiments, for each of the 300 simulated realizations we optimize the a posteriori solution in Gurobi with a two-hour time limit. As Table 1 details, the sample mean of the total dispatch time in the a posteriori solution is within approximately 4% of the operational benchmark. For comparison, operational SDD models are notoriously difficult to benchmark; many works in the SDD literature do not include lower bounds at all, and those that do often report larger gaps against a posteriori solutions even for complex heuristic policies, e.g. [22]. We therefore conclude that our model prescribes reasonable operational behavior, in line with what a dispatcher could accomplish with sophisticated decision-support tools.

### 5.2 Single Vehicle Policy

We now suppose the service region is served with a single delivery vehicle. Since only a single vehicle is available, we also move the cutoff time back to 2:00 PM ($N = 50$), which results in a more reasonable workload. Using Algorithm 1 and the results in Section 3.2 we obtain a two-dispatch solution: The first dispatch serves 35 orders with a time of 204.5 minutes, and the second takes the remaining 15 orders with a
duration of 125.5 minutes.

The operational benchmark for the single-vehicle policy here is similar in spirit to the many vehicle one. As our discussion in Section 3.2 suggests, each dispatch should take all currently unserved orders. To determine the time of first dispatch \( \alpha \), we operationally mimic the equation

\[
\alpha + f(\alpha) + f(N - \alpha) = T,
\]

which determines the dispatch time in the tactical model. As orders arrive, we iteratively solve a TSP for the unserved orders and track \( \tau \), the sum of setup time, service time for accumulated orders, and optimal tour duration. We dispatch the vehicle at the time \( \alpha \) satisfying

\[
\alpha + \tau + f(N - \alpha) = T.
\]

While the vehicle is en route, we know its return time and thus the maximum possible duration of the next dispatch such that it returns by \( T \). As new orders arrive, we again solve the TSP for these locations, and accept orders only until the next dispatch’s total duration matches the remaining time in the service day. As in the many vehicle case, this corresponds to an operational dynamic adjustment of the order cutoff time that, in expectation, should match \( N \) if our model is accurate. Note also that we can extend the benchmark in an analogous fashion to single vehicle problems with more than two dispatches.

We again simulated 300 realizations of the operational benchmark, and report results in Table 2 in a similar fashion to Table 1. As in the previous experiment, our tactical model predicts the expected number of orders served and the expected total routing time of the operational benchmark to within 1% or less. Furthermore, we again implemented an a posteriori benchmark, which allows the dispatcher to optimize the two delivery routes with full advance knowledge of the time and location of each order, using the same experimental setup as in the many-vehicle case. In this case, the operational policy is within approximately 13% of the a posteriori benchmark, suggesting again that our system is modeling reasonable behavior when compared to what a complex operational decision support tool can hope for. Interestingly, in the single-vehicle case, we observe that the a posteriori solution moves more orders to the second dispatch; with advance knowledge of future order locations, the dispatcher is able to anticipate areas where more orders will take place late in the day, and wait until the second dispatch to serve these locations more efficiently. Nevertheless, the gaps we observe between the operational benchmark and a posteriori solution in this experiment are still in line with other results in the operational SDD literature [22].

<table>
<thead>
<tr>
<th></th>
<th>Tactical</th>
<th>Operational</th>
<th>A Posteriori</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Dispatch Quantity</td>
<td>35.01 units</td>
<td>34.96 units (± 0.30)</td>
<td>12.99 units (± 1.32)</td>
</tr>
<tr>
<td>First Dispatch Time</td>
<td>204.52 min.</td>
<td>203.45 min. (± 0.74)</td>
<td>87.28 min. (± 6.48)</td>
</tr>
<tr>
<td>Second Dispatch Quantity</td>
<td>14.99 units</td>
<td>15.57 units (± 0.63)</td>
<td>37.54 units (± 1.45)</td>
</tr>
<tr>
<td>Second Dispatch Time</td>
<td>125.41 min.</td>
<td>123.73 min. (± 3.27)</td>
<td>196.91 min. (± 6.61)</td>
</tr>
<tr>
<td>Total Quantity</td>
<td>50.00 units</td>
<td>50.53 units (± 0.73)</td>
<td>50.53 units (± 0.73)</td>
</tr>
<tr>
<td>Total Time</td>
<td>329.93 min.</td>
<td>327.18 min. (± 3.79)</td>
<td>284.19 min. (± 3.26)</td>
</tr>
</tbody>
</table>

Table 2: Computational study results, one vehicle policy.

6 Conclusions

We have proposed a tactical analysis model for same-day delivery that captures operations at the level of a single depot and its service region. By approximating the order arrival process and the delivery vehicle routing time, we are able to derive simple and transparent optimal solutions for the model that describe the average performance of a reasonable SDD system; our empirical validation shows that the model can indeed predict system behavior very accurately at an operational level.

Using our model, a system manager can easily perform what-if analysis on various potential system configurations, and compare the cost and operating conditions of these configurations to decide various tactical
questions, such as the size of the delivery fleet, the order cutoff time, or whether to have vehicles deliver to the entire service region versus partitioning the region by vehicle. We similarly hope the community derives other applications of the model in SDD tactical design.

Our results motivate several interesting avenues for research. One possibility is to further investigate the interplay of service region partitioning with our model. For example, it would be useful for SDD managers to know precisely when partitioning is preferable to serving the whole region, or to determine if the system can operate more efficiently by serving different parts of the service region differently. A manager may wish to offer SDD with different cutoff times in different areas, based on how efficiently customers in the different areas can be served; perhaps more densely populated urban centers can be profitably served until later in the day while outlying suburban areas need an earlier cutoff. More generally, it may be useful to address partitioning and fleet sizing in tandem, where some sub-regions are served by more vehicles because of higher order density, while others get a smaller delivery fleet because of relative order paucity.

References


7 Appendix

7.1 Proof of Theorem 1

The approach is a proof by contradiction: If some optimal dispatch policy does not have the properties listed in the theorem, we show that a desired policy can be constructed.

Fix any non-trivial (strictly positive dispatch sizes) optimal set of dispatches as \{ (t^*_d, q^*_d, i^*_d) \}_{d=1}^{D^*}. Without loss of generality we can assume that each dispatch in this solution is made with a unique vehicle. Additionally, we can assume that vehicles do not idly wait at the depot after they have enough orders to dispatch, that is, \( t^*_d = q^*_d \) and \( t^*_{d-1} + q^*_{d} \) for \( d = 2, 3, ..., D^* \). Because of the property that \( f(\cdot) \) is increasing in the number of orders served we can w.l.o.g. assume that consecutive dispatch sizes are non-increasing. Say this was untrue for a pair of dispatches, namely \( (t^*_d, q^*_d, i^*_d) \), and \( (t^*_{d+1}, q^*_{d+1}, i^*_{d+1}) \). Then from the above assumptions it must be true that \( t^*_{d+1} - t^*_d = q^*_{d+1} \). Thus we can “swap” these dispatches and arrive at a new feasible solution of equal cost: \( (t^*_d + q^*_{d+1} - q^*_{d}, q^*_{d+1}, i^*_{d+1}) \), and \( (t^*_{d+1}, q^*_{d+1}, i^*_{d+1}) \). Repeating this process will induce the desired ordering on the consecutive dispatch sizes.

With our above assumptions on the fixed optimal dispatch policy, we now claim that the first \( D^* - 1 \) dispatches arrive back at the depot at exactly time \( t = T \). If this claim is true, then each of the first \( D^* - 1 \) dispatch vehicles will take all of the realized orders waiting at the depot at the time of dispatch and return at exactly time \( t = T \), thus proving that the MVP is optimal.

For the sake of contradiction, assume there exists at least one of the first \( D^* - 1 \) dispatches in the optimal solution such that the corresponding vehicle arrives back to the depot before time \( T \). Observe the first such dispatch, indexed by \( d' \), and the subsequent dispatch \( d' + 1 \). These dispatches serve \( q^*_{d'} \) and
There must exist an \( \varepsilon > 0 \) such that the dispatches \((t^*_d + \varepsilon, q^*_d + \varepsilon, i^*_d)\), and \((t^*_{d'+1}, q^*_{d'+1} - \varepsilon, i^*_{d'+1})\) are feasible, which follows from the property that \( f(\cdot) \) is concave and increasing in the number of orders served. Furthermore, since \( f(q^*_d + \varepsilon) + f(q^*_{d'+1} - \varepsilon) \leq f(q^*_d) + f(q^*_{d'+1}) \), via Lemma 3, we can create a feasible dispatch solution where the \( d' \) dispatch arrives back at a later time with cost less than or equal to the assumed optimal policy. One can choose \( \varepsilon \) large enough such that the arrival time will be exactly \( t = T \), with the exception of the case where it is feasible to choose \( \varepsilon = q^*_{d'+1} \), that is, \( t^*_d + q^*_d + f(q^*_d + q^*_{d'+1}) \leq T \), then set \( \varepsilon = q^*_{d'+1} \), remove the \( d' + 1 \) dispatch, and then repeat the argument. Thus, either the \( d' \) dispatch will now arrive back at \( t = T \), and we can re-order the remaining dispatches and repeat the argument with one less dispatch in question, or we can remove the \( d' + 1 \) dispatch entirely, and then repeat the argument. Therefore, by finite iteration, we can transform our previous optimal solution into one where all of the non-last dispatches arrive back at the depot at exactly time \( t = T \) or we can contradict that this policy was optimal to begin with.

Therefore the MVP is an optimal dispatch policy. Clearly, any problem that needs to be solved with fleet size less than \( D^* \) cannot possibly return better objective value than the MVP over \( D^* \) vehicles.

7.2 Proof of Lemma 4

The approach is to split the proof into cases. One can continually dispatch a vehicle with exactly \( q_{\text{min}} \) orders until a final dispatch can feasibly leave with all unserved orders.

Assume we are given an SDD problem with a singleton fleet, which satisfies the Minimal Dispatch Size and Guaranteed Feasibility conditions. Let \( D' = \text{floor}(N/q_{\text{min}}) \).

Assume \( D' \leq 1 \). Observe the dispatch policy \((t_1 = N, q_1 = N, 1)\). Note that \( D' \leq 1 \) implies that \( N < 2q_{\text{min}} \). Because \( f(\cdot) \) is increasing we have \( f(N) < f(2q_{\text{min}}) \). Because \( N + f(N) < N + f(2q_{\text{min}}) \leq T \), we see that the final dispatch returns to the depot by the end of the service day, and the lemma is proven for \( D' \leq 1 \).
Assume \( D' \geq 2 \). Observe the dispatch policy \((t_1 = q_{\text{min}}, q_1 = q_{\text{min}}, 1), (t_2 = 2q_{\text{min}}, q_2 = q_{\text{min}}, 1), \ldots, (t_{D' - 1} = (D' - 1)q_{\text{min}}, q_{D' - 1} = q_{\text{min}}, 1), (t_{D'} = N, q_{D'} = N - (D' - 1)q_{\text{min}}, 1)\). By construction, it must be the case that \( f(q_{\text{min}}) \leq q_{\text{min}} \). Thus, all of the first \( D' - 1 \) dispatches return to the depot before or just as \( q_{\text{min}} \) orders accumulate. Additionally by choice of \( D' \) it must be the case the \( q_{D'} \geq q_{\text{min}} \). Thus, the first \( D' - 1 \) dispatches all return to the depot before the next dispatch must leave. The last dispatch takes all of the remaining orders and thus it remains to be seen that the last dispatch will return by time \( t = T \). By the choice of \( D' \) we have the \( 2q_{\text{min}} > q_{D'} \). Because \( f(\cdot) \) is increasing we have \( f(q_{D'}) < f(2q_{\text{min}}) \). Because \( N + f(q_{D'}) < N + f(2q_{\text{min}}) \leq T \), we see that the final dispatch returns to the depot by the end of the service day, and the lemma is proven for \( D' \geq 2 \).

### 7.3 Proof of Theorem 5

The argument is a proof by contradiction: We first show that two of the properties of an optimal dispatch policy can be easily attained. Then we show that the final property can be changed by successive changes that preserve optimality.

Assume we are given an SDD problem with a singleton fleet, which satisfies the Minimal Dispatch Size and Guaranteed Feasibility conditions. By Lemma \[ \alpha \] the problem is feasible. If the policy \((t_1 = N, q_1 = N, i_1 = 1)\) is feasible, that is, \( N + f(N) \leq T \), then we are done. So, assume any feasible dispatch policy requires two or more dispatches from the vehicle. Therefore, we can fix an optimal policy, \( P_1 \), as \( \{(t_d^*, q_d^*, t_d^*)\}_{d=1}^{D^*} \) where \( D^* \geq 2 \).

In the case that \( P_1 \) violates (2) or (3), simply push all of the idle waiting time before the first dispatch. Additionally, push any remaining time at the end of the service day to be before the first dispatch. This new policy will still be feasible, and therefore be cost optimal. Furthermore, this transformed policy will now satisfy (2) and (3). Fix this policy, \( P_2 \), as \( \{(t_d^*, q_d^*, t_d^*)\}_{d=1}^{D^{**}} \).

Now we show that \( P_2 \) can be transformed into an optimal dispatch policy such that consecutive dispatch sizes are non-increasing. Assume that this is not already the case. Compute the smallest index, \( d' \), such that \( q_d^{**} < q_{d'+1}^{**} \), and observe the dispatches \((t_d^{**}, q_d^{**}, 1)\) and \((t_{d'+1}^{**}, q_{d'+1}^{**}, 1)\). Because \( d' < D^{**} \), we know that \( q_d^{**} \geq q_{\text{min}} \) by the Minimum Dispatch Size condition. It follows that \( f(q_d^{**}) < q_d^{**} \). Because \( t_{d'+1}^{**} = t_d^{**} + f(q_d^{**}) \) (no idle waiting), it must be the case that \( t_{d'+1}^{**} > t_d^{**} \). This means that at the time of the \( d' \) dispatch the number of realized orders at the depot decreases by \( q_d^{**} \), and does not increase by more than \( q_d^{**} \) orders by the time of the \( d' + 1 \) dispatch, when \( q_{d'+1}^{**} \) orders are delivered. Thus, it must be the case that there were enough realized orders at the depot to send \( q_d^{**} \) orders for delivery at time \( t_d^{**} \). This implies that the dispatches, \((t_{d'}^{**}, q_{d'}^{**}, 1)\) and \((t_{d'}^{**} + f(q_{d'}^{**}), q_{d'}^{**}, 1)\), are feasible. This process can be iterated until all consecutive dispatch sizes are non-increasing; denote this policy as \( P_3 \), or \( \{(t_d^{***}, q_d^{***}, t_d^{***})\}_{d=1}^{D^{***}} \). Note that the re-ordering from \( P_2 \) to \( P_3 \) implies that \( P_3 \) also defines an optimal dispatch policy where (2) and (3) remain satisfied.

If (1) is satisfied for \( P_3 \), then we are done. Now assume (1) remains unsatisfied. Observe the first dispatch in which not all of realized orders at the depot are taken for delivery, indexed by \( d' \), and its consecutive dispatch \( d' + 1 \). Let \( \varepsilon > 0 \) represent the amount of orders left at the depot when the \( d' \) dispatch occurs. Also define \( \sigma = \varepsilon \) in the case that \( d' + 1 = D^{***} \), and equal to \( \min\{\varepsilon, q_{d'+1}^{***} - q_{\text{min}}\} \), otherwise. From here, it is feasible to use dispatches of \((t_d^{***}, q_d^{***} + \sigma, 1)\) and \((t_{d'}^{***} + f(q_{d'}^{***} + \sigma), q_{d'}^{***} + 1 - \sigma, 1)\) instead of \((t_d^{***}, q_d^{***}, 1)\) and \((t_{d'}^{***} + f(q_{d'}^{***}), q_{d'}^{***} + 1, 1)\). Name this re-balanced dispatch policy \( P_4 \). By Lemma \[ \beta \] \( P_4 \) defines an optimal dispatch policy where (2) and (3) remain satisfied or we have contradicted the optimality of \( P_3 \). If \( \sigma = \varepsilon \), then \( P_4 \) is an optimal dispatch policy where all of the first \( d' \) dispatches leave the depot with all of the realized orders, and we can repeat the re-balancing argument with less dispatches in question for a re-ordered \( P_4 \) (P5). In the case that \( \varepsilon > \sigma = q_{d'+1}^{***} - q_{\text{min}} \) and \( d' + 1 < D^{***} \) then we can use the arguments above to re-order the dispatch sizes in \( P_4 \) to be non-increasing. We can continually re-balance the \( d' \) dispatch with a newly ordered \( d' + 1 \) dispatch until either a) dispatch \( d' \) dispatches will all of the realized orders at the
depot and the re-balancing argument repeats with less dispatches in question, or b) all of the dispatches after $d'$ (there will be at least 2) are of size less than or equal to $q_{\text{min}}$, so we are no longer allowed to re-balance dispatch sizes.

To summarize, the only way for (1) to remain unsatisfied for an optimal policy, P6 or \( \{ (d_{d_1}^6, q_{d_1}^6, t_{d_1}^6) \}_{d_1=1}^{d_{6^*}} \), which satisfies (2) and (3), is in the case that there is some first dispatch, indexed by \( m \) and with objective value of the policy using \( D_f \) for the last dispatch which may be smaller.

As the heuristic uses \( D_f \) in many cases, at some point we must either satisfy (2) and (3), or we must create an order that is entirely not on the depot. Thus we have for P6, that \( r_{D^*_6}^f < N \leq t_{D^*_6}^f \), then we can consolidate the orders to a single dispatch of \( (N, q_{D^*_6}^f + q_{D^*_6}^f, 1) \) producing a transformed policy, P7. By the Guaranteed Feasibility condition, P7 is feasible as \( q_{D^*_6}^f + q_{D^*_6}^f \leq 2q_{\text{min}} \). By Lemma [3] it must be true that \( f(q_{D^*_6}^f + q_{D^*_6}^f) \leq f(q_{D^*_6}^f) + f(q_{D^*_6}^f) \). If this inequality is strict then we violate P6 being optimal (contradiction). If this inequality is an equality then (3) must have been untrue of P6 (contradiction). Thus, it must be for P6 that \( N \leq t_{D^*_6}^f \leq r_{D^*_6}^f \).

Thus we have for P6, that \( N \leq t_{D^*_6}^f \leq r_{D^*_6}^f \). We can consolidate the orders to a single dispatch of \( (t_{D^*_6}^f, q_{D^*_6}^f + q_{D^*_6}^f, 1) \) producing a transformed policy, P8. By Lemma [3] P8 defines an optimal dispatch policy where (2) and (3) remain satisfied or we have contradicted the optimality of P6. We can derive another optimal policy, P9, by re-ordering the dispatch sizes of P8 to be non-increasing. Note that P9 also satisfies (2) and (3), and has one less dispatch than P6.

It must be the case for P9 that either the $d'$ dispatch serves strictly more orders than the $d'$ dispatch of P6, or the $d'$ dispatch serves the same number of orders as it did in P6, but the $d'+1$ dispatch serves strictly more than $q_{\text{min}}$ orders. Thus we can continuously re-balance, consolidate, and re-order until we arrive at a dispatch policy which satisfies (2) and (3) and the first $d'$ orders all serve all of the realized orders at the depot at the time of dispatch, or until the $d'$ dispatch has become the last dispatch as all of the later dispatches have been consolidated and re-balanced into the $d'$ dispatch. Thus we continue the process until (1) is also satisfied and we are done.

7.4 Proof of Theorem[7]

As the heuristic uses $m - 1 + D$ dispatches in total, the many vehicle policy must also use no more than $m - 1 + D$ vehicles. Let $z_{\text{many}}$ represent the objective value of the many vehicle policy, $z_{m-1+D}$ represent the objective value of the policy using $m - 1 + D$ vehicles and $z_{m}$ represent the objective value of the optimal policy where $m$ vehicles are used. Finally, let $z_{\text{HMD}}$ represent the objective value of the heuristic policy, using $m$ vehicles where the last vehicle is dispatched $D$ times. Thus, we know that $z_{\text{many}} = z_{m-1+D} \leq z_{m} \leq z_{\text{HMD}}$ and our objective is to show that $z_{\text{HMD}} \leq z_{m-1+D} \leq z_{\text{many}}$.

Let $z_{\text{many}}^1$ represent the total routing cost in the many vehicle policy of using the first $m - 1$ vehicles, while $z_{\text{many}}^2$ represents the remaining cost from the $m$-th dispatch onward. Similarly, let $z_{\text{HMD}}^1$ represent the total routing cost in the heuristic of using the first $m - 1$ vehicles, while $z_{\text{HMD}}^2$ represents the remaining cost from the last vehicle, which dispatches $D$ times from the depot. We have that $z_{\text{many}} = z_{\text{HMD}}^1$ by construction of the heuristic. If we assume that some $N_1$ orders remain to be dispatched after each policy identically dispatches the first $m - 1$ vehicles, then by Lemma [7] the worst thing to do with these remaining orders with $D$ dispatches is to serve them all at equal size, $\frac{N_1}{D}$, which has a total cost of $Df(\frac{N_1}{D})$. The most cost effective policy for routing the remaining orders would be to send them all at once at a cost of $f(N_1)$. Therefore, we can deduce that $f(N_1) \leq z_{\text{many}}^1 \leq z_{\text{HMD}}^1 \leq Df(\frac{N_1}{D})$. As we have $f(n) = bn + c\sqrt{n}$, it follows that $Df(\frac{N_1}{D}) \leq \sqrt{D}f(N_1)$. This implies that $z_{\text{HMD}}^2 \leq \sqrt{D}z_{\text{many}}^2$. 

24
Figure 8: Visual representation of the heuristic with $m = 3$ and $D = 3$. The first two dispatches account for the cost in $\mathcal{H}_mD$ while the last three dispatches account for the cost in $z^H_mD$

Figure 9: Visual representation of many vehicle policy over the same problem instance as in Figure 8. Note that only 4 vehicles are used. The first two dispatches account for the cost in $\mathcal{m}^1$ while the last two dispatches account for the cost in $\mathcal{m}^2$.

It remains to compare $\mathcal{m}^1$ and $\mathcal{m}^HmD$. Because the dispatch sizes are non-increasing in the MVP, we can deduce that the cost of every one of the first $m - 1$ dispatches, which are summed to equal $\mathcal{m}^1$, is greater than or equal to that of any of the costs in the remaining dispatches, which are summed to equal $\mathcal{m}^2$. Thus we can deduce that $\mathcal{m}^1 \geq \frac{m - 1}{m - 1 + D} \mathcal{m}^2$ and $\mathcal{m}^2 \leq \frac{D}{m - 1 + D} \mathcal{m}^1$.

Combining the two, $\mathcal{m}^HmD = \mathcal{m}^1 + \mathcal{m}^2 \leq \mathcal{m}^1 + \sqrt{D}\mathcal{m}^2 = \mathcal{m} + (\sqrt{D} - 1)\mathcal{m}^2 \leq \mathcal{m} (1 + \frac{D(\sqrt{D} - 1)}{m - 1 + D}) = \mathcal{m} \frac{(m - 1 + D\sqrt{D})}{m - 1 + D} \leq \mathcal{m} \frac{(m - 1 + D\sqrt{D})}{m - 1 + D}$, as desired.

7.5 Proof of Proposition 12

We split this proof into cases. Assume we are given a $U$ such that $0 \leq U \leq T - f(2q_{\min})$ and $i_{\max}$ is less than or equal to 2. Suppose $U = 0$, then trivially we have $N^{\ast} = 0$. So, assume that $U > 0$, and therefore $2 \geq i_{\max} \geq 1$.

Assume that $i_{\max} = 1$. Then the range $0 \leq N \leq U$ can be partitioned as $N \in \{0\} \cup (0, U]$. We know from Theorem 5 that given $N \in (0, U]$, the optimal dispatch policy is given by a single dispatch of size $N$ at cost $f(N)$. Therefore, $\pi(N)$, is a convex function over the interval $(0, U]$. Thus, either $N = 0$, or $N = U$ will maximize the profit function.

Now assume that $i_{\max} = 2$. Then the range $0 \leq N \leq U$ can be partitioned as $N \in \{0\} \cup (0, \bar{N}_1] \cup (\bar{N}_1, U]$. We know from Theorem 5 that given $N \in (\bar{N}_1, U]$, the optimal dispatch policy can be fully described by the time of first departure $\alpha_N$. Additionally, we know $g(N) = T - \alpha_N$, and $\alpha_N + f(\alpha_N) + f(N - \alpha_N) = T$. Which means we can write $N = \alpha_N + f^{-1}(T - \alpha_N - f(\alpha_N))$. Thus $N$ can be written as a convex function of $\alpha_N$, and thus $g(N)$ is a concave function with respect to $N$. Thus $\pi(N)$ is a convex function over the interval
\( \tilde{N}_1, U \). From before we also have that \( \pi(N) \) is a convex function over the interval \( (0, \tilde{N}_1) \). Thus the solution to the profit maximization function can be found at \( N = 0, N = \tilde{N}_1 \), or \( N = U \). Thus, the claim is proven.

### 7.5.1 A Posteriori Formulations

We use the following integer programming formulations to compute a posteriori solutions in our computational study.

#### Parameters

- Node set: order locations \( L = \{1, \ldots, n\} \), depot 0
- Arc set: ordered pairs of nodes, \( a = (i, j) \).
- Travel time: \( \tau_{ij}, i, j \in L \cup 0 \), includes depot setup time and order service time as necessary
- Release time: \( r_i \geq 0, i \in L \), the time when order \( i \) is ready
- Deadline: \( T \)
- Fleet size or number of routes: \( K \) (\( K = 2 \) in our experiments)

#### Decision Variables

- \( x^k_{ij} \): indicates if vehicle/route \( k \) goes from \( i \) to \( j \)
- \( d_k \): departure time of vehicle/route \( k \)

The many-vehicle formulation is then given by

\[
\begin{align*}
\min_{d, x} & \sum_{k=1}^{K} \sum_{i,j} \tau_{ij} x^k_{ij} \\
\text{s.t.} & \sum_{a \in \delta^+(i)} x^k_{a} = \sum_{a \in \delta^-(i)} x^k_{a}, & i \in L \cup 0, & k = 1, \ldots, K \\
\sum_{k=1}^{K} \sum_{a \in \delta^+(i)} x^k_{a} &= 1, & i \in L \\
\sum_{a \subseteq S} x^k_{a} &\leq |S| - 1, & S \subseteq L, & k = 1, \ldots, K \\
d_k + \sum_{i,j} \tau_{ij} x^k_{ij} &\leq T, & k = 1, \ldots, K \\
d_k - r_i \sum_{a \in \delta^+(i)} x^k_{a} &\geq 0, & i \in L, & k = 1, \ldots, K \\
d_k &\geq 0, & x^k_{ij} \in \{0, 1\}. & (3g)
\end{align*}
\]

In the formulation, (3b) ensures flow balance; (3c) requires each order to be served by a vehicle; (3d) eliminates subtours; (3e) establishes a route duration limit, so each vehicle returns by \( T \); (3f) prevents a vehicle serving \( i \) from departing the depot before the order is ready.
In the single-vehicle case, assume routes are indexed in order of departure. For all routes except \( K \), we replace (3e) with

\[
d_k - d_{k+1} + \sum_{i,j} \tau_{ij} x^k_{ij} \leq 0, \quad k = 1, \ldots, K - 1.
\]  
(4)

This ensures route \( k \) finishes before \( k + 1 \) begins, so one vehicle can perform all routes.