The Dynamic Dispatch Waves Problem for Same-Day Delivery

Mathias A. Klapp[1], Alan L. Erera[2], Alejandro Toriello[2]

[1]Engineering School, Pontificia Universidad Católica de Chile
Santiago, Chile
maklapp *at* ing.puc *dot* cl

[2]H. Milton Stewart School of Industrial and Systems Engieering, Georgia Institute of Technology
Atlanta, Georgia 30332
{aerera, atoriello} *at* isye *dot* gatech *dot* edu

May 23, 2017

Abstract

We study same-day delivery systems by formulating the Dynamic Dispatch Waves Problem (DDWP), which models an order dispatching problem faced by a distribution center, where orders arise dynamically throughout a service day and must be delivered by day's end. At each decision epoch (wave), the system's operator chooses whether or not to dispatch a single vehicle loaded with orders ready for service, to minimize vehicle travel and penalties for unserved requests. We formulate an arc-based integer programming model and design local search heuristics to solve a deterministic DDWP where order arrival times are known in advance, use this variant to design an *a priori* solution approach, and provide two approaches to obtain dynamic policies using the *a priori* solution. We test and compare solution approaches on two sets of instances with different settings of geography, size, information dynamism, and order timing variability. The computational results suggest that our best dynamic policy can reduce the average cost of an *a priori* policy by 9.1% and substantially improves the fraction of orders delivered (order coverage), demonstrating the importance of reactive optimization for dynamic short-deadline delivery services. We also analyze the tradeoff between two common SDD objectives: total cost minimization versus order coverage maximization. We find structural differences in the dispatch frequency and route duration of solutions for the two different objectives, and demonstrate empirically that small increases in order coverage may require substantial increases in vehicle travel cost.

1 Introduction

Same-day delivery (SDD) is increasingly being offered by retailers and logistics service providers to expand e-commerce. The internet sector represented over 7% of the U.S. retail industry and was expected to grow 14% annually in 2015 [19, 30]. We define SDD as a distribution service that prepares, dispatches and delivers orders to the customer's location on the same day the order from the customer is placed. Amazon, a provider of SDD, has implemented the service in more than 25 U.S. metropolitan areas as of October 2016, and has also piloted "Prime Now," an even faster one-hour delivery service. These services are designed to satisfy demand for instant gratification when ordering consumer products and at the same time to discourage physical store visits [27, 41]. Like all urban delivery operations, same-day delivery operations can be costly due to small order sizes to be delivered to many geographically-dispersed locations, and they are additionally challenged by less time to consolidate orders into effective vehicle routes.

A core logistics process within any delivery system is order distribution from stocking locations to delivery locations. Distribution decisions can be divided into dispatch decisions which select the dispatch times and the orders to be delivered in each delivery vehicle trip, and routing decisions which sequence the delivery locations for each dispatch. Dispatch time selection for same-day services is often more challenging than traditional distribution, since vehicles may not be dispatched simultaneously at the beginning of the operating day and may also be reused during the day. At a dispatch time, it may also be sensible in a same-day service to not send some ready orders if there is reasonable likelihood that later (unknown) orders may arrive that consolidate well with them. In this research, we will explicitly consider these challenges. We will only consider problems where dispatch times are selected from a fixed and finite number of candidates, both for simplicity and to maximize compatibility with warehouse processes such as order picking.

In [27], we defined the Dynamic Dispatch Waves Problem (DDWP) for a simplified SDD distribution system with delivery locations on the one-dimensional line. The DDWP is an order delivery problem with dynamic dispatch and routing decisions for a single vehicle during a fixed-duration operating period (*i.e.*, a day) partitioned in W dispatch *waves*. Each wave is a time point (decision epoch) when picking and packing of a set of orders is completed, and a vehicle (if available at the depot) can be loaded and dispatched from the depot to serve a subset of open delivery orders, or wait for the next wave. Open orders are defined as those ready to be dispatched and not previously served. After the vehicle completes a dispatch route, it returns to the depot and is ready to be dispatched again. At each wave, complete information is known for all open orders, and probabilistic information is available describing potential future orders. The objective is to minimize vehicle operating costs and penalties for open orders that remain unserved at the end of the operating period.

The DDWP captures two fundamental and important tradeoffs in same-day distribution. First, there is a tradeoff between waiting and dispatching a vehicle at a wave. When a vehicle is dispatched, the set of open orders is reduced, but the opportunity to observe and serve future orders near ones in the current route is lost. Conversely, when the vehicle is not dispatched, the time remaining to serve the set of open and future orders is reduced. Second, there is a tradeoff between dispatching a long route that serves many orders versus a short ones with fewer orders. The former consumes more time and keeps the vehicle away from the depot longer, but requires less time per customer visited due to routing economies. A shorter route requires more time per customer, but enables the vehicle to be reused sooner.

1.1 Contributions

In this paper, we formulate the single vehicle DDWP in a general network topology, and provide a better model of same-day delivery operations in a typical road network. We capture fundamental tradeoffs in dynamic dispatch decision-making, and model a canonical prize-collecting version of the DDWP. We consider the following to be our primary contributions.

1. We formulate a MIP model for a deterministic variant where all order arrival times are known in advance, which we leverage to provide **lower bounds** for the stochastic-dynamic case via information relaxation and simulation.

2. We use the deterministic model to find an **optimal *a priori* solution** to the stochastic variant, by showing that the *a priori* optimization problem is equivalent to an instance of the deterministic variant with an expanded customer set and adjusted penalties. To our knowledge, this is the first time an optimal *a priori* policy is presented for a model that includes route sequencing decisions in the same-day delivery distribution literature. We also design construction and local search heuristics based on the prize-collecting Traveling Salesman Problem to complement commercial MIP solvers and speed up the identification of solutions to problem instances.

3. We develop two approaches to obtain **dynamic policies** using the *a priori* model. The first uses a rollout scheme to dispatch according to an updated *a priori* solution, and the latter relies instead on fast heuristic modifications to the initial *a priori* solution.

4. We prove that the expected cost of the rollout policy is no higher than the optimal *a priori* policy

cost. We later confirm the benefits of dynamic policies with computational experiments. Experiments show that the percentage reductions in optimality gap and operating cost from an *a priori* policy to a dynamic rollout policy average 47.6% and 9.1%, respectively, while the order fill rate improves by 4.6%. The marginal benefits of dynamic policies increase for instances with greater order arrival variability and less information disclosed before the start of the operation. Performance measures for dynamic policies also exhibit less variability across instances.

5. Finally, we analyze the tradeoff between two common objectives in SDD: minimizing total costs (including vehicle travel time) versus maximizing order coverage in an empirical study. One might suppose that these two objectives would lead to similar solutions, since low-cost routes create more available vehicle time to serve additional orders. In contrast, we instead find fundamentally different solution structures for the two cases in terms of number of vehicles dispatched, route length and initial wait at the depot. Results indicate that one should expect significant sacrifices in vehicle routing efficiency in order to maximize the order fill rate, and the cost of an additional customer covered becomes more expensive as order coverage increases.

The remainder of the paper is organized as follows. A literature review is presented in Section 1.2. Section 2 defines notation and formulates the DDWP, Section 3 covers the deterministic problem, and Sections 4 and 5 respectively cover *a priori* and dynamic policies. Finally, Section 6 outlines the results of a computational study, and we conclude with Section 7.

1.2 Literature Review

The DDWP can be classified within the broad family of stochastic vehicle routing problems (VRP) that are extensions of the deterministic VRP [22, 38]. The simplest models for this problem family are *a priori* optimization models [15, 18, 21, 26], where an initial solution is designed before the operation starts and then pre-defined recourse rules, *i.e.*, simple plan corrections, are used to (potentially) modify it during operation as information is disclosed. More complex models for dynamic-stochastic VRPs are dynamic policies that adapt to revealed information during the operating period, and allow for re-optimization of structural routing and scheduling decisions; see [12, 29, 31, 36]. A restricted case of dynamic policies are *a priori* rollout policies that proved succesful for the VRP with stochastic demands [23, 24]. Such approaches recompute an *a priori* policy at each decision epoch for the remaining operating period using the current system state as starting point.

A problem related to the DDWP is the VRP with probabilistic customer arrivals (PC-VRP) where customer orders arrive over time with uncertainty; see [16, 20, 26, 28, 40] for examples of *a priori* models and [2, 3, 4, 5, 12, 42] for dynamic models for such problems. Our problem differs from those considered in this literature, which are designed primarily for pick-up operations. In pick-up operations, a vehicle route may be modified after dispatch, both by re-routing currently planned deliveries and adding new customers. In contrast, our order delivery model assumes customized delivery orders; it is not possible to add new customers to a route that has been previously dispatched without returning the vehicle to the depot.

Another closely-related problem is the Dynamic Multiperiod Routing Problem (DMPRP) [2], which models a depot with a fleet of vehicles executing delivery operations over a time horizon partitioned into days. Dynamic requests with service time windows are disclosed over time, and each day the decision maker chooses which customers to serve or postpone, and the corresponding vehicle routes. This model assumes daily decision epochs, and that a vehicle is always available for dispatch at each epoch. In SDD operations, it may be sensible to dispatch a vehicle on a route that spans multiple decision epochs; in this case, if a vehicle is dispatched at time t on a route of duration x to serve a set of orders, it cannot be dispatched again until a time after its return to the depot at $t+x$.

A deterministic order delivery problem related to the DDWP is the VRP with release dates (VRP-rd) [7, 17] that considers dispatching orders with known release times. A feasible solution of the VRP-rd serves all orders. In addition to ignoring the stochastic and dynamic nature of customer delivery orders, this problem does not consider the decision to reject requests, which is important when a system operates with a fixed fleet of vehicles and delivery deadlines.

Perhaps the closest problem in the literature to the DDWP is the Same Day Delivery Problem for Online Sales (SDDP) [41], which considers an order delivery model with dynamic request arrival and a fleet of vehicles, each performing multiple trips per day. However, there are fundamental differences in assumptions and solution methodologies between this setting and the DDWP. The model assumes narrow request service time windows, *i.e.*, one or two hours within a 10-hour day operating period; these tight time constraints correspond to rapid delivery settings, like Amazon's "Prime Now" service. Serving customers is set as objective, while ignoring vehicle travel costs. Solutions are found by adapting a scenario-based-planing heuristic (see [12]) which relies in part on simulation. The work of [39] extends on the SDDP by allowing preemptive returns to the depot to pick-up newly realized demand before delivering all packages currently loaded in the vehicle. Besides practical implementation issues, our preliminary computational evidence suggests that preemptive returns many only offer small marginal benefits, especially if alternatively returns are planned before each vehicle dispatch based on available probabilistic information.

Another set of related work is presented in [8, 10], where a deterministic VRP model with time windows having multiple delivery routes per vehicle is developed. Medium and large sized instances are solved with an adaptive large neighborhood search (ALNS) heuristic. The optimization model objective is to maximize the number of served orders and travel costs in hierarchical order, while multiple trips per vehicle are induced by the inclusion of a route duration limit; experiments show that this constraint leads to short routes, with an average of four customers served per route. In [9], the authors extend this model to a dynamic setting where online requests arrive and the decision maker instantaneously accepts or rejects them for service. Each time a new request arrives, acceptance decisions are made based on an insertion policy using scenario-based planning heuristics.

2 DDWP problem formulation

We formally define the DDWP as a Markov Decision Process (MDP); see the text [33] for a reference on MDPs. We start by describing the notation and elements of our model:

1. **Operating period.** Let $\mathscr{W} := \{1, \ldots, W\}$ be the set of *waves* (decision epochs) and let ℓ be the time interval between two consecutive waves. The number $w \in \mathscr{W}$ represents the "waves-to-go" before $w = 0$, the deadline for the vehicle to finish all deliveries and return to the depot.

2. **Customer orders and geography.** Let $N := \{1, \ldots, n\}$ be the set of all *potential* customer orders. Since each potential order is associated with a delivery location, let N and a single deport ($i = 0$) be the nodes of a complete, undirected graph $\mathscr{G} = (N \cup \{0\}, E)$, where E is the set of edges. We assume that a delivery vehicle requires t_e time and incurs cost c_e to traverse an edge $e \in E$, and for simplicity that time and cost values are proportional to each other ($c_e = \gamma t_e$), non-negative, and satisfy

the triangle inequality.

3. **Order ready times.** Each order $i \in N$ is ready for dispatch at a random wave $\tau_i \in \mathscr{W}$, drawn from an order-dependent distribution. It is possible that order i does not arrive at all; we represent this case by setting $\tau_i = -1$, therefore $\tau_i \in \mathscr{W} \cup \{-1\}$. We assume that ready times for different orders are independent, and that the initial set of open orders $\{i \in N : \tau_i \geq w\}$ is known at wave w.

4. **Penalties.** Each order i has a penalty $\beta_i > 0$, to be paid if the order arrives and is not served by a vehicle by $w = 0$.

We now are ready to formulate an MDP model for the DDWP. If the vehicle is available at the depot at wave w, the system state is (w, R, P), where $R \subseteq N$ is the set of open orders and $P \subseteq N$ is the set of remaining *potential* orders with an unknown random arrival time ($\tau_i < w$). An order i cannot be simultaneously open and potential, so that R and P must be disjoint. Orders in $N \setminus \{R \cup P\}$ are *closed*, meaning that they were already served before wave w. The maximum number of waves W and three possible states for each order (open-closed-potential) define an $\mathscr{O}(3^n W)$ bound on the cardinality of the state space.

An action at any non-terminal state $(w, R, P) : w \geq 1$ is defined as a vehicle dispatch that serves a subset of open orders $S \subseteq R$; $S = \emptyset$ represents is the decision for the vehicle to wait at the depot. A vehicle dispatch action for a non-empty S implies that the vehicle is away from the depot for $t(S)$ time, where $t(S)$ is the minimum time required by any tour to visit S and return to the depot (a Traveling Salesman Problem (TSP) tour visiting nodes $S \cup \{0\}$); define $t(\emptyset) = 0$. Given decision S at wave w, the vehicle is not available for a next possible dispatch until wave $w_S^- := w - \max\{1, \lceil t(S)/\ell \rceil\}$. Note furthermore for feasibility that S must be constrained such that $w_S^- \geq 0$, to ensure the vehicle returns to the depot by the end of the day.

All potential subsets of R imply $\mathscr{O}(2^n)$ possible actions. Selecting an action S in state (w, R, P) produces a transition to state $\left(w_S^-, R \setminus S \cup F_w^{w_S^-}, P \setminus F_w^{w_S^-}\right)$ where $F_w^{w'} := \{i \in N : w > \tau_i \geq w'\}$ is the random set of newly arriving orders at waves $w-1, \ldots, w'$, and $R \setminus S$ is the set of orders in R left open by action S.

Let $C_w(R, P)$ be a set function representing the minimum expected cost-to-go at state (w, R, P), so that $C_W\left(\hat{R}, N \setminus \hat{R}\right)$ is the minimum expected cost at the start of the operation, given an initial set of open orders $\hat{R}$. The dynamic program defined in equation set (1) computes $C_W\left(\hat{R}, N \setminus \hat{R}\right)$ recursively over w. The cost $C_0(R, P)$ of terminal state $(0, R, P)$ is equal to the sum of penalties of unserved open orders R, and subsequently, for each $w \in \mathscr{W}$ the cost-to-go at state (w, R, P) is equal to the minimum cost of the wait action or any feasible dispatch action. Define an optimal action as a subset $S_w(R,P) \subseteq R$ that attains $C_w(R,P)$ at a given state (w,R,P), and an optimal policy as a vector of optimal actions for each possible state of the system:

$$C_0(R,P) = \textstyle\sum_{i \in R} \beta_i, \qquad \forall (R,P) \in \Xi \tag{1a}$$

$$C_w(R,P) = \min_{S \subseteq R : w_S^- \geq 0} \left\{ \gamma t(S) + \mathbb{E}_{F_w^{w_S^-}} \left[C_{w_S^-} \left(R \setminus S \cup F_w^{w_S^-}, P \setminus F_w^{w_S^-} \right) \right] \right\}, \qquad \forall w \in \mathscr{W}, \forall (R,P) \in \Xi. \tag{1b}$$

Finding an optimal policy directly with (1) is difficult; the state space is exponential in n ($\mathscr{O}(3^n W)$), the action space for each state is exponential in n ($\mathscr{O}(2^n)$), and the expectation expressions contain an exponential number of terms ($\mathscr{O}(2^n)$). Moreover, simply evaluating the cost-to-go for a state-action pair is NP-hard, since computing $t(S)$ is the TSP optimization problem on $S \cup \{0\}$. Given these difficulties, we focus on finding good heuristic policies for the DDWP.

3 The Deterministic DDWP

Suppose the wave τ_i at which order $i \in N$ is ready for dispatch is known at the beginning of the operating horizon. Then, the sets of orders ready at each wave $w \in \mathscr{W}$ are known in advance, but it remains infeasible to serve an order i with a vehicle dispatch prior to τ_i. Let $N_w := \{i \in N : a_i \leq w \leq \tau_i\}$ be the set of orders ready and feasible to serve at wave $w \in \mathscr{W}$, where a_i defines the latest wave to feasibly serve order i, *i.e.*, $a_i := \lceil 2t_{\{0,i\}}/\ell \rceil$. Problem 3.1 states the deterministic DDWP.

Problem 3.1 (Deterministic DDWP). *Find a collection of vehicle dispatches represented by mutually disjoint subsets of orders $\{S_w \subseteq N_w : w \in \mathscr{W}\}$ that minimize $\sum_{w \in \mathscr{W}} \{t(S_w) - \sum_{i \in S_w} \beta_i\}$, where the dispatches S_w for each wave $w \in \mathscr{W}$ satisfy the following two conditions:*

- *Returns before the end of the day:* $w_{S_w}^- \geq 0$
- *Does not overlap another vehicle dispatch in time: if $S_w \neq \emptyset$, then $S_v = \emptyset$ for all $v \in \{w_{S_w}^-, \ldots, w-1\}$.*

Property 3.2 is originally shown for the simpler deterministic DDWP with delivery locations on the half-line extending from the depot in [27], and its shifting arguments can be trivially generalized to the problem in this paper.

Property 3.2 (No wait after a dispatch)**.** *There exists an optimal solution to Problem 3.1 in which the vehicle does not wait after the first dispatch has occurred.*

We now formulate the deterministic DDWP as an Integer Program (IP). Define $E_w := \{e \in E, a_e \leq w \leq b_e\} \subset E$ for each $w \in \mathscr{W}$ as the set of feasible edges for a vehicle dispatch at wave w, where $a_{\{i,j\}} = \lceil (t_{\{0,i\}} + t_{\{i,j\}} + t_{\{0,j\}}) / \ell \rceil$ and $b_{\{i,j\}} = \min\{\tau_i, \tau_j\}$, for each $\{i,j\} \in E$. Also, define the cut set $E_w(V) = \{\{i,j\} \in E_w : i \in V, j \notin V\}$, for any subset $V \subseteq (N_w \cup \{0\})$. Problem 3.1 is equivalent to the IP

$$C^*(\tau) = \min_{\{\mathbf{x},\mathbf{y},\mathbf{v},\mathbf{z}\}} \sum_{i \in N : \tau_i > 0} \beta_i \left(1 - \sum_{w=a_i}^{\tau_i} y_i^w\right) + \gamma \sum_{w \in \mathscr{W}} \sum_{e \in E_w} t_e x_e^w \tag{2a}$$

$$\text{s.t.} \sum_{w=a_i}^{\tau_i} y_i^w \leq 1, \qquad \forall i \in N \tag{2b}$$

$$\sum_{e \in E_w(\{0\})} x_e^w \leq 2, \qquad \forall w \in \mathscr{W} \tag{2c}$$

$$\sum_{e \in E_w(S)} x_e^w \geq 2 y_i^w, \qquad \forall w \in \mathscr{W}, \forall S \subseteq N_w, \forall i \in S \tag{2d}$$

$$\sum_{e \in E_w} t_e x_e^w \leq \ell \sum_{k<w} (w-k) v_k^w, \qquad \forall w \in \mathscr{W} \tag{2e}$$

$$\sum_{k<W} z_k + \sum_{k<W} v_k^W = 1 \tag{2f}$$

$$\sum_{k<w} v_k^w = \sum_{k>w} v_w^k + z_w, \qquad \forall w \in \mathscr{W} \setminus \{W\} \tag{2g}$$

$$v_k^w \in \{0,1\}, \qquad \forall w \in \mathscr{W}, \forall k \in \mathscr{W} \cup \{0\} : k < w \tag{2h}$$

$$z_k \in \{0,1\}, \qquad \forall k \in \mathscr{W} \cup \{0\} : k < W \tag{2i}$$

$$y_i^w \in \{0,1\}, \forall i \in N_w, \text{ and } x_e^w \in \{0,1,2\}, \forall e \in E_w. \qquad \forall w \in \mathscr{W} \tag{2j}$$

Variable y_i^w is equal to 1 if a dispatch at wave w serves order i, and 0 otherwise; x_e^w is equal to $m \in \{0,1,2\}$ if the vehicle traverses edge e m times during a dispatch at wave w, and 0 otherwise; v_k^w is equal to 1 if a dispatch at w returns at wave k, and 0 otherwise; and z_k is equal to 1 if the vehicle waits at the depot until wave k, and 0 otherwise ($z_0 = 1$ implies no dispatch throughout the horizon).

Constraints (2b) guarantee serving each order i exactly once at wave w ($y_i^w = 1$) or, alternatively, leaving it unserved ($y_i^w = 0,\ w = a_i, \dots, \tau_i$). Constraints (2c) - (2d) guarantee that vector $\mathbf{x}^w = \{x_e^w : e \in E_w\}$ defines a feasible tour only visiting orders selected by $\{y_i^w : i \in N_w\}$. Constraints (2e) force routes to satisfy duration limits determined by v_k^w. Finally, constraints (2f)-(2g) enforce vehicle conservation throughout time.

We solve instances of (2) using a standard Branch & Cut approach equipped with dynamic generation for subtour elimination cuts based on approaches for the TSP; see [6]. We start with singleton constraints (2d), *i.e.*, $S = \{i\}$. If we find an integer vector $\mathbf{x}^w$ for wave w at any given node in the Branch & Bound tree, we check if it has a subtour in $\mathscr{O}(n)$ time and add the corresponding cut. Alternatively, if $\mathbf{x}^w$ is fractional, we check if it violates 2-connectivity by solving a Minimum Cut problem, and if so we add the corresponding cut from (2d) and repeat.

Consider again the stochastic DDWP defined in Section 2. We can estimate a lower bound on the optimal expected cost of (1) using a Perfect Information Relaxation (PIR) cost $C_{PIR} = \mathbb{E}_\tau(C^*(\tau))$, computed by disregarding the "non-anticipative" dynamics of the problem and solving for the optimal value $C^*(\tau)$ for each possible realization τ [14, 34]. To estimate the PIR, we create a random sample of M realizations, $\{\tau^m\}$, find $C^*(\tau^m)$ for each $m \in \{1, \dots, M\}$, and compute the sample average $\sum_{m=1}^{M} C^*(\tau^m)/M$.

4 *A priori* policies

In this section, we develop a procedure to compute an *a priori* policy, in which a schedule specifying the waves at which to dispatch the vehicle and the routes serving ordered subsets of delivery orders (locations) at each dispatch is determined using only information disclosed before the operation begins at wave W. The expected cost of such a policy can be computed easily when schedules and routes are not modified during operations. In this case, the only uncertain costs are associated with penalties for unserved orders. Let $\mathbb{P}(\tau_i = w)$ be the probability that order i becomes ready first at wave w. To simplify notation, let these probability values include all information regarding the initial set of orders $\hat{R}$, *i.e.*, $\mathbb{P}(\tau_i = W) = 1$ if $i \in \hat{R}$, and 0 otherwise. It can be shown that this *a priori* problem is equivalent to solving a deterministic DDWP instance in which each order $i \in N$ is duplicated W times within its geographic location and each copy is assumed to be ready at each wave $w \in \mathscr{W}$, with an adjusted penalty for not serving the order equal to $\beta_i P(\tau_i = w)$.

We note that there exists an optimal solution to this *a priori* model visiting the same geographic location at most once. To see why, note that a solution without this property can be transformed into another feasible solution with no greater cost by removing all visits to a location except the latest one in time; dispatch and routing costs are not increased, and the same set of orders remain served. This observation allows us to define an IP model for the *a priori* problem where planning to serve order i at wave w indicates that order i is covered if it arrives during any wave $v \in \{w, w+1, \dots, W\}$; this action thus reduces the expected penalty by $\beta_i \mathbb{P}(\tau_i \geq w)$. Define the expected penalty to be paid if no vehicle dispatches are planned as $\beta^0 := \sum_{i \in N} \beta_i \mathbb{P}(\tau_i \geq 1)$, the earliest wave that a vehicle can serve order i as $b_i := \max\{w : \mathbb{P}(\tau_i = w) > 0\}$, and redefine the sets N_w and E_w accordingly. Then, the *a priori* DDWP is given by

$$C^*_{AP}(\hat{R}) = \min_{\{\mathbf{x},\mathbf{y},\mathbf{v},\mathbf{z}\} \in \text{(2c) - (2j)}} \beta^0 - \sum_{i \in N} \sum_{w=a_i}^{b_i} \beta_i \mathbb{P}(\tau_i \geq w) y_i^w + \gamma \sum_{w \in \mathscr{W}} \sum_{e \in E_w} t_e x_e^w \tag{3a}$$

$$\text{s.t.} \sum_{w=a_i}^{b_i} y_i^w \leq 1, \qquad \forall i \in N, \tag{3b}$$

Note that (3) has a similar feasible region to (2), but now with an expected value objective function. When the solution does not serve order i, we pay a penalty discounted by the arrival probability $\mathbb{P}(\tau_i \geq 1)$, and when the solution serves i at wave w, we pay the penalty discounted by the probability of a later arrival, $\mathbb{P}(1 \leq \tau_i < w)$.

We can improve the performance of the *a priori* policy during execution by skipping planned visits to orders that are not ready on time from a route dispatched at wave w; the triangle inequality guarantees this can only reduce costs. We define by $\mathscr{Q}_w(\mathbf{x}^\mathbf{w})$ the expected duration of a dispatch at wave w given by $\mathbf{x}^w$; [26] provides a closed form expression to compute this expected cost for a given route in $\mathscr{O}(n^2)$ time. The expected cost of such a policy for a given *a priori* feasible solution is equal to

$$\beta^0 - \sum_{i \in N} \sum_{w=a_i}^{b_i} \beta_i \mathbb{P}(\tau_i \geq w) y_i^w + \sum_{w \in \mathscr{W}} \mathscr{Q}_w(\mathbf{x}^\mathbf{w}). \tag{4}$$

We could design an *a priori* solution that considers order-skipping recourse proactively when planning a solution with an extension of the L-shaped method described for the Probabilistic Traveling Salesman Problem (PTSP) in [28]. We implemented and tested this approach but were only able to solve small instances to optimality ($n = 25$, $W = 3$). Moreover, the benefits in cost savings over (3) were small (under

2%), and we postulate that such small savings are intuitive. If arrival probabilities are high, the probabilistic routing cost in the objective collapses to a deterministic routing cost, similar to the PTSP. However, this is a prize-collecting problem, which is fundamentally different from the PTSP, and if the arrival probabilities are small, the probabilistic routing cost must be balanced with the expected penalty cost in the objective. Thus, it becomes more important to make good order dispatch selections and less important to route effectively. We leave the design of efficient heuristic procedures that minimize (4) to future research efforts; see *e.g.*, [13, 35] for related work.

4.1 Practical considerations when solving the *a priori* model

We propose an improvement local search heuristic (LS) that exploits structure of the *a priori* model and complements a MIP solver. Specifically, we use this heuristic in two phases when solving problem (3). First, we run LS for each new feasible solution s identified during the branch-and-bound tree search, and update the incumbent if LS produces a solution with lower cost. Second, we use the heuristic during a solution construction phase to generate a good initial feasible solution for the MIP solver.

Our LS procedure uses three separate neighborhood searches given solution s: (1) intra-route local search (IntraLS), *i.e.*, single route node selection and re-sequencing; (2) inter-route local search (InterLS), *i.e.*, node exchanges between routes and re-sequencing; and (3) dispatch profile local search (DPLS), *i.e.*, search over the subsets of dispatch waves selected by the operation. Each search procedure is described in Appendix 8.1. We execute the three-level search until no improving solution is found.

5 Dynamic Policies

A priori policies, particularly when adjusted via recourse actions, may yield reasonable solutions to many problems. However, even in the one-dimensional case [27], there exist instance families for which an optimal *a priori* policy with recourse is arbitrarily worse than an optimal dynamic policy, *i.e.*, $C^*_{AP}(\hat{R})/C^*(\hat{R}) \to \infty$. With this motivation, we develop dynamic policies.

5.1 *A Priori*-Based Rollout Policy

A natural dynamic policy is to roll out the *a priori* policy [23, 24]. At each wave $w \in \mathscr{W}$ when the vehicle is available at the depot, we recompute an optimal *a priori* policy beginning at wave w using the current system state (w, R, P) to define a new, reduced problem over the remaining operating period. If the policy decides at wave w to dispatch a route serving customer set $S \subset R$, then this decision is implemented and a new *a priori* policy is computed again at wave w_S^-; otherwise, a new *a priori* policy is computed at $w-1$ after waiting for one wave.

In Proposition 5.1, we claim that roll out of the *a priori* optimal solution guarantees nonnegative expected cost savings. The proof is provided in Appendix 8.2, and is based on results in [25].

Proposition 5.1. *Let $\hat{R}$ be the initial set of open orders available at $w = W$, let $C_{AP}(\hat{R})$ be the optimal value of the a priori problem in Equations* (3) *and let $C_{RP}(\hat{R})$ be the expected cost of our policy that rolls out the a priori optimal plan. Then $C_{RP}(\hat{R}) \leq C_{AP}(\hat{R})$.*

Computing such a rollout policy requires the solution of $\mathscr{O}(W)$ problems, which is computationally expensive for larger instances and may be impractical. However, there are multiple ways to improve computational performance. First, one can warmstart the solution of the *a priori* problem at wave w using the most recently computed solution (from wave $q > w$), adjusted to skip all planned orders that are not ready yet and then improved via the LS procedure. It is also not necessary to solve each *a priori* problem to optimality.

A more substantial simplification is to begin the rollout process only at the first wave where a dispatch is planned in the initial *a priori* solution (computed at wave W); we call this approach the *restricted* version of the rollout policy. In preliminary experiments the restricted version performed similarly to a full rollout policy in terms of expected cost, indicating that the AP policy is adequately choosing an initial dispatch. We find this to be a useful insight for managers: assuming average behavior in the future appears to be sufficient when making an initial dispatch decision. Conversely, once the AP policy recommends an initial dispatch, a rollout policy performs better by incorporating newly arrived information.

5.2 Greedy *a priori*-based Policy

We also propose simpler rollout strategies that do not rely on repeatedly solving the *a priori* problem. One such approach takes advantage of the relationship between the DDWP and the prize-collecting TSP (PCTSP). To initiate the approach, we solve the *a priori* problem at initial wave W. We then use this solution to guide a rollout procedure that only solves PCTSP problems as follows. At each wave $w < W$ for which the *a priori* solution dictates a vehicle dispatch, we compute a maximum duration of the route by ensuring that the vehicle returns to the depot by the next planned dispatch wave in the *a priori* solution (or by the deadline). Then, we determine which open orders to serve and in which sequence by solving a PC-TSP, where the input set of orders are all open orders that are not planned in the *a priori* solution for a later dispatch wave. Appendix 8.3 describes this heuristic algorithm in detail.

This greedy policy only computes a solution to minimize expected cost once, at the beginning of the horizon. Although the dispatch waves are not updated, the policy dynamically adjusts when orders that were not planned to be served arrive, when orders arrive later than their planned service wave, and when re-routing can improve transport time and cost. This last feature alone enables this approach to outperform the simpler *a priori* policy with order-skipping recourse.

6 Computational Experiments

We now present a set of computational experiments designed using a family of randomly generated instances with the objective of testing the quality of our heuristic policies, and to get qualitative insights regarding the management of vehicle dispatches in a same-day delivery context. Table 1 summarizes the heuristic policies computed for each instance. All heuristics were programmed in Java and computed using one thread of a Xeon E5620 processor with up to 12Gb RAM, using CPLEX 12.6 when necessary as a MIP solver.

Table 1: Heuristic policies computed in our experiments

symbol	strategy	section
AP	*a priori* policy + order-skipping	4
GP	Greedy *a priori*-based policy	5.2
RP	Rollout of *a priori* policy (restricted version)	5.1

6.1 Design of data sets

We generated 240 data sets to evaluate our policies over different performance indicators. Each data set has a specific geography of 50 orders, a known subset $\hat{R} \subseteq \{1,\ldots,50\}$ of orders ready at the start of the operating period, and a vector of ready wave probabilities for orders with unknown arrival waves.

The geography for each data set is defined by a random seed $g \in \{0,\ldots,4\}$ used to assign 50 different locations over a 51×51 square subset of $\mathbb{R}^2$ following a discrete uniform distribution $U(0,50)$ for each component of the location's coordinate and with the depot located at coordinate $(25,25)$. We ruled out repeated coordinates to have a more interesting geography. Travel times between locations are specified by the ℓ_1-norm (Manhattan distance) between two locations, chosen to roughly model urban travel times. All data sets share a common horizon with $W = 6$ possible dispatch waves, each with duration $\ell = 100$ time units. The duration of a round-trip visiting any single order is less than or equal to 100 time units, and thus can be completed in a single dispatch wave; this implies that a single penalty can always be avoided.

For each potential order $i \in \{1,..,50\}$, its ready wave τ_i is a discrete random variable, independently distributed with probability $\mathbb{P}(\tau_i = W) = p_{start}$ of being ready at the start of the operating period (the order's degree of dynamism); a conditional probability $\mathbb{P}(\tau_i = -1|\tau_i < W) = p_{out}$ of not arriving during the operation period; and a conditional discrete uniform distribution with probability $\mathbb{P}(\tau_i = w|1 \leq \tau_i < W) = (\min(W-1,\mu_i+\sigma) - \max(1,\mu_i-\sigma)+1)^{-1}$ of arriving during the operation at any wave $w \in \{\max(1,\mu_i-\sigma),\ldots,\min(W-1,\mu_i+\sigma)\}$. The parameter μ_i represents the mean ready wave and is drawn for each i with equal probability between 1 and $W-1$, and σ represents variability. Each data set uses a triple $(\sigma, p_{start}, p_{out}) : \sigma \in \{\text{Lo} = 1, \text{Hi} = 6\}$, $p_{start} \in \{10\%, 15\%, 25\%, 50\%\}$, $p_{out} \in \{20\%, 40\%\}$, and a setting of the seed $h \in \{0,1,2\}$ to draw the set $\hat{R}$ of orders ready at the start.

We created $M = 50$ realizations of ready waves for each data set using the probabilistic model above. These scenarios are used as a common sample to estimate a perfect information bound and the expected cost of all policies. Each data set has a unique value of $(g, \sigma, p_{start}, p_{out}, h)$, and all sets, including their simulated realizations, are online at `sites.google.com/site/maklappor/ddwp-data-sets`.

6.2 Set 1: Base experiments

For our first set of experiments, we build 3 instances by considering the first 25, 35 and 50 orders for each one of the 240 data sets. This makes a set of 720 instances with 3 different problem sizes. The penalty for leaving a realized order i unattended is set as the duration of a round-trip to order i from the depot, $\beta_i := 2t_{\{0,i\}}$; this setting roughly models a system that serves all realized orders by outsourcing the service of each order that remains open at the end of the day to a direct delivery service. Given this penalty structure, there is always potential service profitability; in particular, there is an economic incentive to dispatch the vehicle in the last wave if any order is open.

We compute the following metrics for each policy over each realization from each instance:

- Total cost ($cost$),
- gap: the percentage increase of the policy's cost over the perfect information bound,
- fill rate (fr): the percentage of orders served by the vehicle over all realized orders,
- $duration/order$: the routes' travel time over the number of served orders,
- $nRoutes$: number of vehicle dispatches,
- $nWaves$: average dispatch length in waves used by each route,
- $iWait$ and $pWait$: number of waves spent waiting at the depot before/after the initial dispatch,
- $time_{off}$: "off-line" solution time, *i.e.*, before the solution is implemented,
- $time_{on}$: total "on-line" solution time during the operating period divided by the number of active decision epochs, *i.e.*, the number of waves in which the policy makes a dispatch decision (which excludes all predetermined waits established by the initial *a priori* policy and dispatch waves jumped by the vehicle routes).

These metrics are averaged for each instance over all $M = 50$ realizations. Table 2 presents average results for each heuristic policy over all instances.

On average, the AP policy's cost is 23.1% higher than the perfect information bound and, as expected, the AP rollout (RP) significantly reduces the average gap by 47.7%, with an average cost reduction of 9.1%.

Table 2: Average results of heuristic policies

metric policy	**AP**	**GP**	**RP**
cost (decrease from AP %)	555	523 (5.8%)	505 (9.1%)
gap	23.1%	16.1%	12.1%
fr	81.6%	85.0%	86.2%
$duration/order$	11.0	11.2	11.2
nRoutes	2.49	2.47	2.55
nWaves	1.43	1.43	1.38
iWait	2.58	2.60	2.65
pWait	0.003	0.005	0.002
$time_{off}$		836.2s.	
$time_{on}$	0.0001s.	0.08s.	85.8s.

The greedy heuristic GP achieves 63% of that cost reduction, suggesting that a simple but dynamic policy can capture a significant amount of the benefits of dynamism.

Also, RP increases the order fill rate over AP by 5.6% by redesigning the solution at each decision moment, which is crucial for logistics service providers interested in providing a better customer service; GP achieves 75.2% of this fill rate increase. This is a tradeoff wherein dynamic policies significantly improve order service, while incurring small increases in duration per order. We also see that dynamic policies slightly increase the average number of dispatches; refer to Appendix 8.4 for more detailed analysis of solution dispatch structure. Finally, note that it is rare for the vehicle to wait after its first dispatch (*pWait*); it is better to keep the vehicle busy serving more orders after an initial wait for consolidation.

All policies require identical offline computing time ($time_{off}$), but differ dramatically in on-line solution time: AP and GP are almost instantaneous online policies, while RP requires more computational power.

In Figure 1, we observe the distribution of the gap over all instances for each heuristic policy. We observe that RP not only outperforms AP on average, but is also less variable in solution quality, with 3.77% deviation versus 7.43%; GP has an intermediate 4.87% deviation.

In Figure 2 we compare the average gap of our heuristic policies between instances sharing parameters of size n, degree of dynamism p_{start}, probability of not showing up p_{out}, and arrival variability σ. In the first graph on the left we see how the average gap increases with the number of potential orders n; this increase may be related to an increase in problem size and complexity, but also to the tightness of the lower bound. Moreover we see that RP's gap difference over AP increases as n grows, with GP in between. We

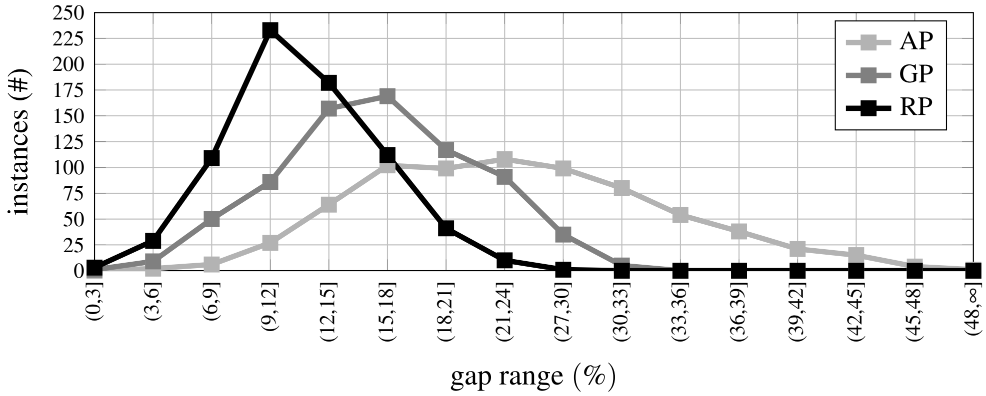


Figure 1: Distribution of gap over all instances

also observe a significant gap reduction for the *a priori* policy (AP) as p_{start} increases. As expected, more information available at the initial wave brings the problem closer to a deterministic problem which allows us to optimize exactly. For dynamic policies, the gap is significantly smaller and tends to be stable over different values of p_{start}, showing the benefit and importance of complex recourse actions when dealing with higher degrees of dynamism. Figure 2 suggests that it may be harder to optimize instances with higher values of p_{out} and σ due to an increase in the problem's uncertainty and/or possibly due to a deterioration of our lower bound. Again, we see that the average gap reduction of dynamic policies over the *a priori* policy is particularly valuable for instances with higher ready time variability.

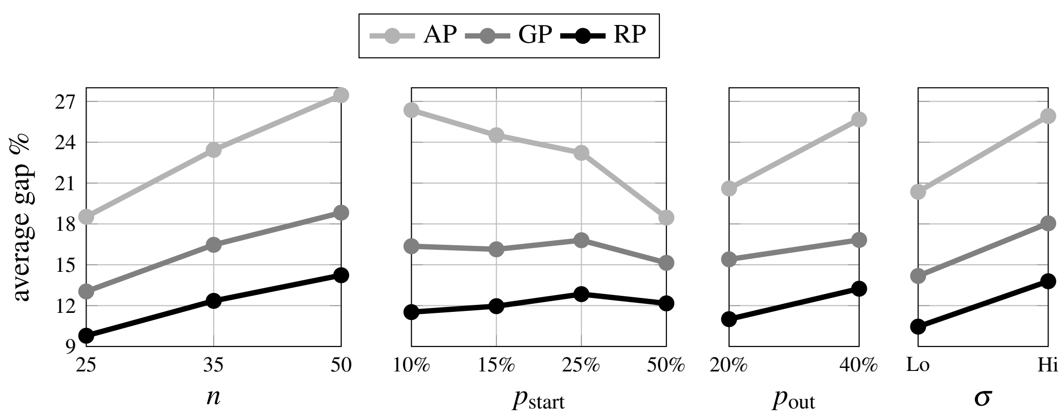


Figure 2: Average gap versus n, p_{start}, p_{out}, and σ

Figure 3 presents the average fill rate as a function of the instance parameters; for reference, maximizing fill rate is the objective function of the model presented in [41]. We see that fr decreases as n increases over all policies, which may indicate congestion related to available vehicle time. Interestingly, this fill rate reduction is marginally decreasing with n, presumably through order consolidation; *i.e.*, at $N = 25$, an increase of 10 customers results in a larger fill rate decrease than an increase of 15 customers at $N = 35$. Also, RP's fill rate is consistently over AP by more than 4.2% regardless of n. The second graph from the left shows how the average fill rate increases by 16.1% for AP and 11.2% for RP when p_{start} increases from 10% to 50%, demonstrating that more information available at the start can significantly increase the number of orders served, especially for the AP policy. The relative difference between AP and RP is higher for instances with higher dynamism, showing again the additional value of dynamic policies. The third graph from the left demonstrates that RP (and dynamic policies more generally) can be useful when opportunities to increase fill rate are presented. While the *a priori* policy's rate remains stable over values of p_{out}, all dynamic policies increase order fill rate, *e.g.*, RP's increases by 1.6%. This may be explained by the fact that RP uses the time gained when initially planned orders do not arrive to serve unexpected and initially unplanned orders that do show up. This effect is also observed when the fill rate is compared to σ.

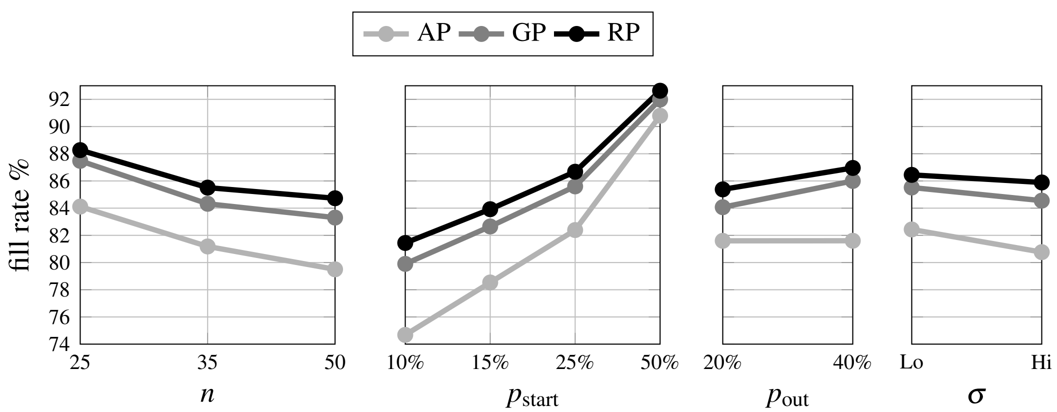


Figure 3: Average fill rate versus n, p_{start}, p_{out}, and σ

The average solution times over all instances disaggregated by number of customers n and the probability p_{start} are presented Table 3. The results on the left table show the off-line solution times shared by all policies. As expected due to the nature of exact MIP models, this time increases exponentially with the number of orders, but should not be problematic since this procedure is intended to run before the start of the operating horizon. In addition, average times increase with p_{start}. The right-hand table shows the average online solution times per dispatch of GP and RP policies. We observe a similar increasing effect over the number of customers.For even larger n, the GP policy is a simpler alternative to consider if RP becomes inefficient; GP still takes less than 1 second per decision and generates better solutions than AP. Another possibility would be to further exploit local search and meta-heuristic procedures; we leave these ideas for future work.

Table 3: Average $time_{on}$ and $time_{off}$ versus p_{start} and n

Average time offline in secs.

$p_{start} \backslash n$	25	35	50
10%	121	338	1689
15%	130	375	1870
25%	224	494	1886
50%	245	579	2084

Average time per decision in secs.

	GP			**RP**		
$p_{start} \backslash n$	25	35	50	25	35	50
0.10	0.06	0.10	0.16	3.4	16.5	147
0.15	0.05	0.12	0.14	4.1	16.1	186
0.25	0.04	0.05	0.07	4.3	32.8	256
0.50	0.03	0.03	0.06	4.7	51.9	305

6.3 Set 2: Route efficiency versus customer service

We now present a second set of computational experiments to study the tradeoff between two plausible objective functions in same-day delivery: minimizing total cost and maximizing (weighted) order coverage; in the DDWP setting this is equivalent to minimizing penalty costs. This coverage goal also matches the objective in the models in [41]. We look for basic tradeoffs, performance of our heuristic policies, and structural differences in our solutions to provide qualitative insights.

We build a new set of 720 instances by making 3 instances for each one of the 240 data sets with 35 orders each. Each instance has a different value of $\alpha \in \{1,2,100\}$ for the penalty setting $\beta_i = 2\alpha d_{0i}$. While the first set ($\alpha = 1$) balances both vehicle traveling costs and penalty costs, the last one ($\alpha = 100$) hierarchically focuses on covering orders first with travel time minimization as a secondary objective; the second set ($\alpha = 2$) is an intermediate case.

Table 4 presents average costs of all heuristic policies and perfect information bound over all instances under different settings of α. It also shows for each dynamic policy the average cost reduction percentage over AP. We first observe that cost reduction percentages of dynamic policies substantially increase with α; this is explained by the order coverage improvement that dynamic policies enjoy, discussed in the first set of experiments. It may also occur because the cost savings from delivering an additional order become

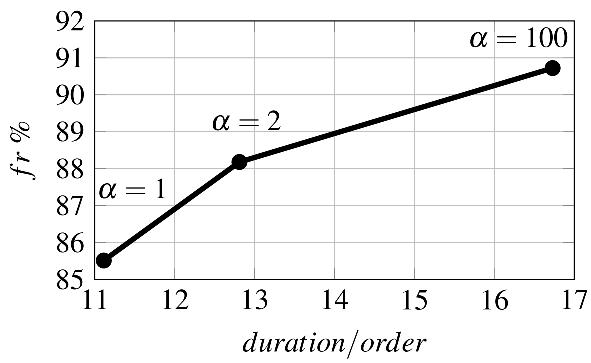
Table 4: Average cost and reduction percentage over AP for each policy under different settings of α

policy\objective	$\alpha = 1$	$\alpha = 2$	$\alpha = 100$
LB	441	546	9395
AP	541	760	18958
GP	511 (5.5%)	711(6.4%)	16402 (13.5%)
RP	493 (8.8%)	670 (11.8%)	13625 (28.1%)

more significant as the weight on penalty costs is bigger. For example, the cost reduction from serving one more order when there are two left unattended is 50%. Second, we observe that RP becomes more attractive compared to GP as the relative importance of penalty costs increases, suggesting that sophisticated dynamic policies that get better fill rate improvements become more valuable as the focus shifts towards coverage.

metric	$\alpha = 1$	$\alpha = 2$	$\alpha = 100$
$duration/order$	11.1	12.8	16.7
fr	85.5%	88.2%	90.7%
$nRoutes$	2.48	3.25	4.77
$nWaves$	1.37	1.23	1.13
$iWait$	2.72	2.14	0.61

(a) Metrics for RP under different settings of α



(b) Pareto chart for RP with fr versus $duration/order$

Figure 4: Average results for RP under different settings of α

The left table in Figure 4 presents results for RP averaged over all instances sharing the same settings of α. The first two rows present the tradeoff between $duration/order$ representing routing efficiency and fr representing order satisfaction. These results are plotted in the graph on the right as a Pareto chart. As expected, we observe that fr increases as order coverage becomes more relevant in the objective, but this fill rate improvement requires a sacrifice in $distance/order$, *i.e.*, the higher the order fill rates, the more inefficient the routes. Moreover, the marginal rate of substitution between fr and $distance/order$ decreases with α. For example, from $\alpha = 1$ to $\alpha = 2$ the average gain in fr per $distance/order$ sacrificed is 1.57%. The same number from $\alpha = 2$ to $\alpha = 100$ is 0.65%, a 59% reduction. For decision makers, this suggests that even in the efficient frontier, the transportation cost of an additional customer covered becomes increasingly more expensive; a cost-focused manager may be willing to sacrifice coverage for routing efficiency at a sufficiently high fill rate.

We further explain the decreasing marginal substitution rate between coverage and routing efficiency by looking at the last three rows of the table in Figure 4. RP creates more opportunities for recourse and reoptimization to improve order coverage; the average number of dispatched routes is increased by 92% and the route duration is reduced by 17%, and the result is reduced routing efficiency. Moreover, RP drastically reduces the initial wait time by 78%, implying that the vehicle is in use for a much larger portion of the day. This is also an interesting insight for managers: when coverage is the objective, vehicles operate longer periods of time with higher maintenance costs and longer workdays for drivers (a higher cost in human resources). Conversely, order consolidation increases when total cost is the objective, routes become longer and efficient, and dispatches are pushed forward in time. This reduces the vehicle operating time, but increases the time between order arrival and dispatch; logistics providers may want to minimize this value to guarantee good service. In Figure 5 we demonstrate this tradeoff graphically, presenting two solutions for the same instance realization. The left solution minimizes total cost while the right one minimizes lost penalties. We clearly see how the right solution has to sacrifice roughly 50% in routing efficiency to increase its order coverage by one order. It also makes 3 more vehicle dispatches and actively uses the vehicle for 2 more waves. In Appendix 8.5 we present analysis of the tradeoff between customer service and routing efficiency under different values of the data set parameters p_{start}, p_{out} and σ .

7 Conclusions

We have formulated the Dynamic Dispatch Waves Problem (DDWP) on general network topologies to study same-day delivery distribution systems. We propose an IP model to solve a deterministic version of the problem, and use this model to derive an optimal *a priori* solution for the stochastic case.

From the *a priori* solution we derive three heuristic policies that differ in their dynamism. The first policy, AP, is an *a priori* solution with a order-skipping recourse; the second policy is a direct rollout of the *a priori* solution, RP; and GP that is a computationally cheaper alternative to RP that rolls out a prize-collecting TSP guided by an initial *a priori* solution.

We tested these policies in simulated instances under different settings of geography, problem size, and

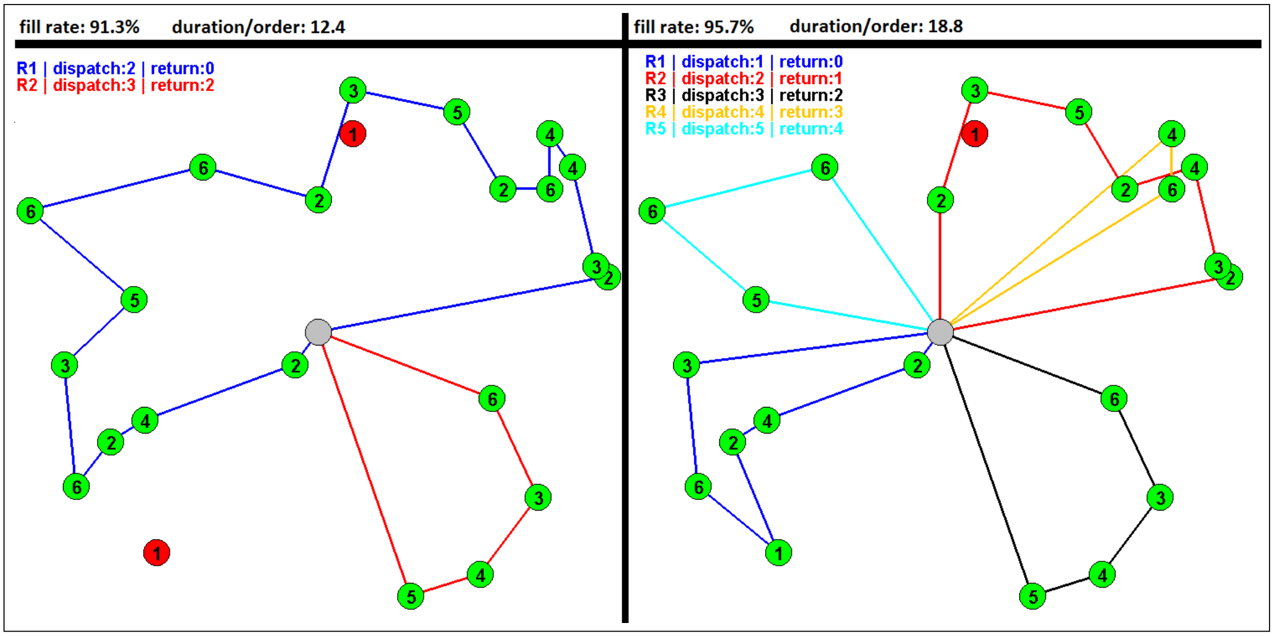


Figure 5: Example of two different solutions to the same instance realization. The left one minimizes total cost and the right one minimizes lost penalties first, then distance cost. Green orders are served and red ones are left unattended. Each order's ready wave is labeled at its node and route dispatch/return waves are shown in the upper-left corner.

order arrival dynamism and variability. The *a priori* policy costs average 23.1% over our perfect information bound. The benefit of a dynamic policy over an *a priori* one can be significant; in our experiments, RP is able to cut AP's cost by 9.1% on average, yielding an average gap of 12.1% over the lower bound and reducing its standard deviation by 3.7%, giving consistently good solutions compared to AP. Its marginal benefit concentrates in improving the order fill rate by 4.6%, which is highly desirable for SDD services. This benefit is achieved by increasing recourse opportunities to catch newly arrived orders, and by sacrificing a small amount of routing efficiency. These benefits are especially significant when the instance dynamism and variability are high; this can be commonplace in SDD systems. We also found little benefit and few occurrences of the vehicle waiting at the depot during the planning horizon once dispatches begin.

RP's computational effort may prevent its deployment (at least using exact optimization) for larger problem sizes, *e.g.*, $n > 50$. In this case, we provide GP that attains on average two-thirds of the cost reduction and fill rate benefit of RP. In general, the DDWP proved quite challenging to solve and future work may consider improvements in heuristic solution of the *a priori* problem. Another interesting question is whether we can exploit probabilistic routing methods to solve the *a priori* problem with recourse.

We also empirically studied the tradeoff between two possible SDD objectives, minimizing total cost and maximizing the weighted order fill rate. One might think that these two objectives deliver similar results, since well-sequenced routes leave more vehicle time available to cover more orders. However, we found that one should expect significant sacrifices in vehicle routing efficiency (distance cost) in order to maximize fill rate, and that the distance cost of an additional customer covered becomes more expensive as order coverage increases. In our results, an RP solution that maximizes order coverage and one that minimizes costs have a 51% difference in traveled distance per order and a 5.2% difference in order fill rate; also, a solution focusing on coverage makes more dispatches (92% more), has 17% shorter routes on average, and starts dispatching earlier, having on average a 65% longer vehicle utilization throughout the day. This has direct implications on driver salaries and vehicle maintenance costs. We also observed that, as fill rate becomes more important, the average improvement of dynamic policies over *a priori* policies increases.

An extension of this model to multiple vehicles seems natural; having a fleet of vehicles could pool the risk associated with leaving orders unattended and therefore reduce costs, *e.g.*, [1]. In general, same-day delivery offers many new challenges to the logistics research community.

References

[1] A. Ak and A. Erera, *A paired-vehicle recourse strategy for the vehicle-routing problem with stochastic demands*, Transportation science **41** (2007), no. 2, 222–237.

[2] M. Albareda-Sambola, E. Fernández, and G. Laporte, *The dynamic multiperiod vehicle routing problem with probabilistic information*, Computers & Operations Research **48** (2014), no. 0, 31–39.

[3] E. Angelelli, N. Bianchessi, R. Mansini, and M. G. Speranza, *Short term strategies for a dynamic multi-period routing problem*, Transportation Research Part C: Emerging Technologies **17** (2009), no. 2, 106–119.

[4] E. Angelelli, M. Savelsbergh, and M.G. Speranza, *Competitive analysis of a dispatch policy for a dynamic multi-period routing problem*, Operations Research Letters **35** (2007), no. 6, 713–721.

[5] E. Angelelli, M.G. Speranza, and M.W.P. Savelsbergh, *Competitive analysis for dynamic multiperiod uncapacitated routing problems*, Networks **49** (2007), no. 4, 308–317.

[6] D.L. Applegate, R.E. Bixby, V. Chvátal, and W.J. Cook, *The traveling salesman problem: A computational study*, Princeton University Press, Princeton, New Jersey, 2006.

[7] C. Archetti, D. Feillet, and M.G. Speranza, *Complexity of routing problems with release dates*, European Journal of Operational Research **247** (2015), no. 3, 797 – 803.

[8] N. Azi, M. Gendreau, and J.-Y. Potvin, *An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles*, European Journal of Operational Research **202** (2010), no. 3, 756 – 763.

[9] ______, *A dynamic vehicle routing problem with multiple delivery routes*, Annals of Operations Research **199** (2012), no. 1, 103–112.

[10] ______, *An adaptive large neighborhood search for a vehicle routing problem with multiple routes*, Computers & Operations Research **41** (2014), 167 – 173.

[11] E. Balas, *The prize collecting traveling salesman problem*, Networks **19** (1989), no. 6, 621–636.

[12] R. Bent and P. van Hentenryck, *Scenario-based planning for partially dynamic vehicle routing with stochastic customers*, Operations Research **52** (2004), no. 6, 977–987.

[13] L. Bianchi and A.M. Campbell, *Extension of the 2-p-opt and 1-shift algorithms to the heterogeneous probabilistic traveling salesman problem*, European Journal of Operational Research **176** (2007), no. 1, 131 – 144.

[14] D. Brown, J. Smith, and P. Sun, *Information relaxations and duality in stochastic dynamic programs*, Operations research **58** (2010), 785–801.

[15] A.M. Campbell and B. Thomas, *Challenges and advances in a priori routing*, The Vehicle Routing Problem: Latest Advances and New Challenges, Springer, 2008, pp. 123–142.

[16] ______, *Probabilistic traveling salesman problem with deadlines*, Transportation Science **42** (2008), no. 1, 1–21.

[17] D. Cattaruzza, N. Absi, and D Feillet, *The multi-trip vehicle routing problem with time windows and release dates*, Transportation Science **50** (2016), no. 2, 676–693.

[18] J.F. Cordeau, G. Laporte, M. Savelsbergh, and D. Vigo, *Vehicle routing*, Transportation, handbooks in operations research and management science **14** (2006), 367–428.

[19] R. DeNale, X. Liu, and D. Weidenhamer, *U.S. Census Bureau News: quaterly retail E-commerce sales - 3rd quarter 2015*, U.S. Department of Commerce (2015), 1–3.

[20] A. Erera, M. Savelsbergh, and E. Uyar, *Fixed routes with backup vehicles for stochastic vehicle routing problems with time constraints*, Networks **54** (2009), no. 4, 270–283.

[21] M. Gendreau, G. Laporte, and R. Séguin, *Stochastic vehicle routing*, European Journal of Operational Research **88** (1996), no. 1, 3–12.

[22] B.L. Golden, S. Raghavan, and E.A. Wasil (eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer, 2008.

[23] J. C. Goodson, J. W. Ohlmann, and B. W. Thomas, *Rollout policies for dynamic solutions to the multivehicle routing problem with stochastic demand and duration limits*, Operations Research **61** (2013), no. 1, 138–154.

[24] J. C. Goodson, B. W. Thomas, and J. W. Ohlmann, *Restocking-based rollout policies for the vehicle routing problem with stochastic demand and duration limits*, Transportation Science **50** (2016), no. 2, 591–607.

[25] ———, *A rollout algorithm framework for heuristic solutions to finite-horizon stochastic dynamic programs*, European Journal of Operational Research **258** (2017), no. 1, 216 – 229.

[26] P. Jaillet, *A priori solution of a traveling salesman problem in which a random subset of the customers are visited*, Operations Research **36** (1988), no. 6, 929–936.

[27] M. Klapp, A. Erera, and A Toriello, *The one-dimensional dynamic dispatch waves problem*, published online on Transportation Science (2016), 1–14.

[28] G. Laporte, F. Louveaux, and H. Mercure, *A priori optimization of the probabilistic traveling salesman problem*, Operations Research **42** (1994), no. 3, 543–549.

[29] A. Larsen, O.B.G. Madsen, and M.M. Solomon, *Recent developments in dynamic vehicle routing systems*, The Vehicle Routing Problem: Latest Advances and New Challenges, Springer, 2008, pp. 199–218.

[30] Matt Lindner, *Global e-commerce sales set to grow 25% in 2015*, https://www.internetretailer.com (2015).

[31] V. Pillac, M. Gendreau, C. Guéret, and A. Medaglia, *A review of dynamic vehicle routing problems*, European Journal of Operational Research **225** (2013), no. 1, 1–11.

[32] M.L. Pinedo, *Scheduling: theory, algorithms, and systems*, Springer Science & Business Media, 2012.

[33] M. Puterman, *Markov decision processes: discrete stochastic dynamic programming*, John Wiley & Sons, 2009.

[34] N. Secomandi and F. Margot, *Reoptimization approaches for the vehicle-routing problem with stochastic demands*, Operations Research **57** (2009), no. 1, 214–230.

[35] H. Tang and E. Miller-Hooks, *Approximate procedures for probabilistic traveling salesperson problem*, Transportation Research Record: Journal of the Transportation Research Board **1882** (2004), 27–36.

[36] B. Thomas, *Dynamic vehicle routing*, Wiley Encyclopedia of Operations Research and Management Science, John Wiley and Sons, Inc., 2010, pp. 1–11.

[37] P. Toth and D. Vigo, *The granular tabu search and its application to the vehicle-routing problem*, Informs Journal on computing **15** (2003), no. 4, 333–346.

[38] ———, *Vehicle routing: Problems, methods, and applications*, vol. 18, SIAM, 2014.

[39] M. W. Ulmer, B. W. Thomas, and D. C. Mattfeld, *Preemptive depot returns for a dynamic same-day delivery problem*, working paper (2016).

[40] S. Voccia, A.M. Campbell, and B. Thomas, *The probabilistic traveling salesman problem with time windows*, EURO Journal on Transportation and Logistics (2012), 1–19.

[41] S. Voccia, A.M. Campbell, and B.W. Thomas, *The same-day delivery problem for online purchases*, Tippie College of Business, University of Iowa (2015), 1–40.

[42] M. Wen, J.-F. Cordeau, G. Laporte, and J. Larsen, *The dynamic multi-period vehicle routing problem*, Computers & Operations Research **37** (2010), no. 9, 1615–1623.

8 Appendix

8.1 Local search procedures over an *a priori* solution

Let $\{r^s_w, w \in \mathscr{W}_s\}$ be the set of routes of a feasible solution s to (3) indexed over the wave subset $\mathscr{W}_s \subseteq \mathscr{W}$ where these dispatches occur; we refer to $\mathscr{W}_s$ as the dispatch profile of a solution s. Each route r^s_w represents an elementary sequence of order visits starting and ending at the depot. The *a priori* cost c_s is defined by (3a), the duration of each route by d^s_w, and its unserved customer set by $U_s := N \setminus \bigcup_{w\in\mathscr{W}_s} r^s_w$.

Algorithm 1 Local Search Procedure

> **procedure** LOCALSEARCH(Solution s)
>   **loop**
>   **if** ($\neg$INTRALS(s) **and** $\neg$INTERLS(s) **and** $\neg$DPLS(s)) **then return** s.

Our LS procedure is given by Algorithm 1 and is composed by three subroutines: IntraLS, InterLS and DPLS. IntraLS, defined in Algorithm 2, exploits the relation between the DDWP and a prize-collecting Traveling Salesman Problem (PCTSP) (see [11])

$$PCTSP(m, Q, \rho) := \min_{\substack{S\subseteq Q:\\ t(S)\leq m}} \left\{ \gamma t(S) - \sum_{i\in S} \rho_i \right\}, \tag{5}$$

where Q is the set of all potential customers, ρ_i is the prize collected when visiting customer $i \in Q$, and m is a maximum route duration. IntraLS is a best move procedure, where a move is described by re-optimizing one route from s leaving all remaining routes unaltered. The procedure chooses a single route r^s_w from s, and solves a PCTSP over a set of nodes $Q := U_s \cup r^s_w$ defined by all orders left unattended if route r^s_w where removed, a maximum route duration $m = d^s_w\ell$ predefined by the waves left available, and prizes β_i discounted by $\mathbb{P}(\tau_i \geq w)$, for $i \in Q$. Any solution s processed by IntraLS contains only routes w that are optimally sequenced and that cannot be improved by selecting a different subset of orders to service from $U_s \cup r^s_w$.

Algorithm 2 Intra-route LS procedure

> **procedure** INTRALS(Solution s)
> $improved \leftarrow false$ and $s^* \leftarrow s$
>   **repeat**
>   **for** $w \in W_s$ **do**
>   Let s' be a copy of s without route r^s_w
>   Solve PCTSP($d^s_w\ell, U_{s'}, \{\beta_i\mathbb{P}(\tau_i \geq w)\}$) and add optimal route found to s' at wave w
>   **if** ($c_{s'} < c_{s^*}$) **then** $s^* \leftarrow s'$ **and** $improved \leftarrow true$
>   **if** ($c_{s^*} < c_s$) **then** $s \leftarrow s^*$
>   **until** $\neg improved$

InterLS uses best move searches over pairs of routes using neighborhoods inspired by those in [37] for the CVRP: two-edge exchanges between routes, removal and reinsertion of a k-order sequence from one route to another, and order swaps between routes. To implement these ideas, we account for two differences between the CVRP and the DDWP. First, we model the prize-collecting component; a move changes penalty

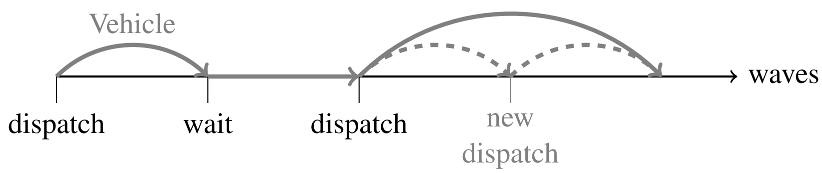


Figure 6: Example of a cut operation where a new dispatch profile is created (dashed flow) from an existing one (continuous flow) by cutting a route into two.

savings (due to the different dispatch time). Second, we check the durations of the new routes to ensure that they remain compatible with the fixed dispatch times of the unchanged routes.

The third neighborhood search that we implement is a Dispatch Profile Local Search (DPLS), described in Algorithm 3. The DPLS search perturbs the structure of the dispatch profile W_s for solution s using five operators: Cut, Merge, Exchange, Start, and Reorder. The first four operators find potential new solutions by solving a PCTSP, while the fifth uses a job scheduling approach.

Algorithm 3 Dispatch Profile Local Search (DPLS)

```
procedure DPLS(Solution s)
    improved ← true
    repeat
        if (¬CUT(s) and ¬MERGE(s) and ¬EXCHANGE(s) and ¬START(s) and ¬REORDER(s)) then
            improved ← false.
    until ¬improved
```

The Cut operator, described in Algorithm 4, searches over all possible dispatch profiles that result when splitting a single dispatch w with duration $d^s_w \geq 2$ into two dispatches with shorter duration; this operator always add an extra return trip to the depot, as depicted in Figure 6. The Merge operator, described in Algorithm 5, works in reverse and searches over all dispatch profiles that arise when merging two consecutive dispatches into a single longer duration dispatch, as shown in Figure 7. The Exchange operator, described in Algorithm 6, changes the dispatch durations of two consecutive dispatches (thus changing the dispatch wave w of the latter); see Figure 8. The Start operator, described in Algorithm 7, searches for a better solution among the (at most) two new dispatch profiles induced by moving the initial dispatch wave of solution s backward or forward one wave, when feasible without altering subsequent dispatches; when the move is backward, the initial dispatch is extended by one wave and when the move is forward, it is reduced by one wave, as depicted in Figure 9.

The Reorder operator reassigns the routes in s to the best possible dispatch waves, without altering the customer visit sequences or the route durations. This assignment problem is equivalent to a single machine job scheduling problem of type $1||\sum_j f_j(w_j)$ (see *e.g.*, [32]): Consider a set of jobs, where each job j corresponds to a route with processing time p_j equal to the route duration (measured in waves). The cost of assigning job j to start wave w is $f_j(w) := \sum_{i\in r^j} \mathbb{P}(\tau_i < w)$. This job scheduling problem is NP-hard, since the single-machine scheduling problem minimizing the weighted sum of tardy jobs can be reduced to this problem. Small instances with fewer than 10 routes can be solved effectively with dynamic programming.

We can in addition enhance the search by calling LS recursively, *i.e.*, we run LS within the Cut, Merge, Exchange, and Start operators on the temporary solution s', after the move gets implemented but before

Algorithm 4 Cut operation over solution s

procedure CUT(Solution s)
    $improved \leftarrow false$ and $s^* \leftarrow s$
    for $w \in W_s$ **do**
        for $v : (w-1) \to (w - d_w^s + 1)$ **do**
            Let s' a copy of s without route r_w^s
            Solve PCTSP($(w-v)\ell, U_{s'}, \{\beta_i \mathbb{P}(\tau_i \geq w)\}$) and add optimal route to s' at wave w.
            Solve PCTSP($(v-w+d_w^s)\ell, U_{s'}, \{\beta_i \mathbb{P}(\tau_i \geq v)\}$) and add optimal route to s' at wave v
            if $(c_{s'} < c_{s^*})$ **then** $s^* \leftarrow s'$ and $improved \leftarrow true$
    if $(c_s > c_{s^*})$ **then** $s \leftarrow s^*$
    return *improved*

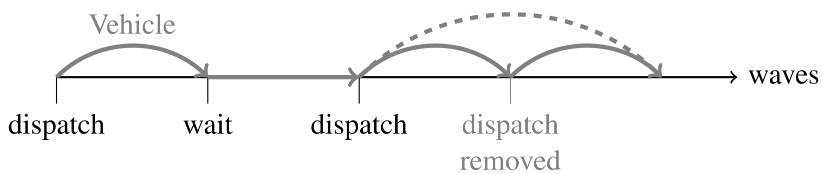


Figure 7: Example of a merge operation where a new dispatch profile is created (dashed flow) from an existing one (continuous flow) by merging two dispatches into one.

Algorithm 5 Merge operation over solution s

procedure MERGE(Solution s)
    $improved \leftarrow false$ and $s^* \leftarrow s$
    for $w \in W_s$ such that $w - d_w^s > 0$ **do**
        Let s' a copy of s without routes r_w^s and $r_{w-d_w^s}^s$
        Solve PCTSP($(d_w^s + d_{w-d_w^s}^s)\ell, U_{s'}, \{\beta_i \mathbb{P}(\tau_i \geq w)\}$) and add optimal route to s' at wave w
        if $(c_{s'} < c_{s^*})$ **then** $s^* \leftarrow s'$ and $improved \leftarrow true$
    if $(c_s > c_{s^*})$ **then** $s \leftarrow s^*$
    return *improved*

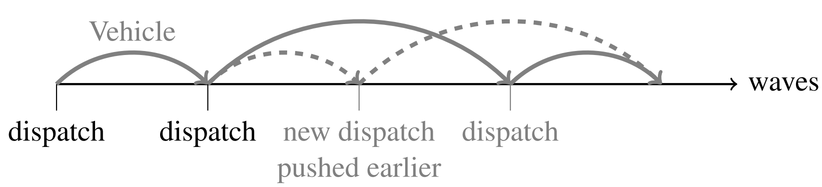


Figure 8: Example of an exchange operation where a dispatch gives one wave to its successor (dashed flow).

Algorithm 6 Exchange operation over solution s

procedure EXCHANGE(Solution s)
  $improved \leftarrow false$ and $s^* \leftarrow s$
  for pair $w_1, w_2 \in W_s$ such that $d^s_{w_1} > 1$ and ($w_1 = w_2 - d^s_{w_2}$ or $w_1 = w_2 + d^s_{w_1}$) **do**
    if ($w_1 > w_2$) **then** Let $w'_1 = w_1$, $w'_2 = w_2 + 1$, $d'_1 = d^s_{w_1} - 1$, and $d'_2 = d^s_{w_2} + 1$.
    else Let $w'_1 = w_2$, $w'_2 = w_1 - 1$, $d'_1 = d^s_{w_2} + 1$, and $d'_2 = d^s_{w_1} - 1$
    Let s' a copy of solution s without routes $r^s_{w_1}$ and $r^s_{w_2}$
    Solve PCTSP($d'_1\ell, U_{s'}, \{\beta_i \mathbb{P}(\tau_i \geq w'_1)\}$) and add optimal route to s' at wave w'_1
    Solve PCTSP($d'_2\ell, U_{s'}, \{\beta_i \mathbb{P}(\tau_i \geq w'_2)\}$) and add optimal route to s' at wave w'_2
    if ($c_{s'} < c_{s^*}$) **then** $s^* \leftarrow s'$ and $improved \leftarrow true$
  if ($c_s > c_{s^*}$) **then** $s \leftarrow s^*$
  return *improved*

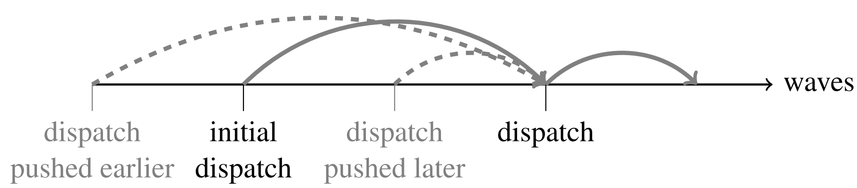


Figure 9: Example of a Start operation where the first dispatch is enlarged/reduced (dashed flow).

Algorithm 7 Start operation over solution s

procedure START(Solution s)
  $improved \leftarrow false$ and $s^* \leftarrow s$
  $ini_s \leftarrow \max\{w \in \mathscr{W}_s\}$
  Let s_1 and s_2 be two copies of solution s without route $r^s_{ini_s}$
  if ($ini_s < W$) **then**
    Solve PCTSP($(d^s_{ini_s} + 1)\ell, U_{s_1}, \{\beta_i \mathbb{P}(\tau_i \geq ini_s + 1)\}$) and add optimal route to s_1 at wave $ini_s + 1$
    if ($c_{s_1} < c_{s^*}$) **then** $s^* \leftarrow s_1$ and $improved \leftarrow true$
  if ($d^s_{ini_s} > 1$) **then**
    Solve PCTSP($(d^s_{ini_s} - 1)\ell, U_{s_2}, \{\beta_i \mathbb{P}(\tau_i \geq ini_s - 1)\}$) and add optimal route to s_2 at wave $ini_s - 1$
    if ($c_{s_2} < c_{s^*}$) **then** $s^* \leftarrow s_2$ and $improved \leftarrow true$
  if ($c_s > c_{s^*}$) **then** $s \leftarrow s^*$
  return *improved*

comparing to the best solution available s^*. In our experiments, we implemented a two-level local search to keep solution times manageable.

In addition to using the LS procedure during the branch-and-bound tree search, we also embed it in a heuristic for building a good initial feasible solution s_0 for the DDWP. This constructive heuristic is given a set of m dispatch profiles, and for each $W_k, k = 1, \ldots, m$ solves a series of sequential PCTSPs over time to build a solution which is then improved with LS. In our experiments, the set of dispatch profiles provided to the construction heuristic was $\{\{1, 2, \ldots, q\}\}$ for all q in $1 \leq q \leq W$, *i.e.*, all possible profiles that include consecutive, single-wave dispatches up to the final wave.

Algorithm 8 Constructive Heuristic

procedure CONSTRUCTIVE(Set of dispatch profiles $\{\mathscr{W}_1, \mathscr{W}_2, \ldots, \mathscr{W}_{s_m}\}$)
    Initialize s^0 as the empty solution
    for $k : 1 \to m$ **do**
        Build $\{d_w, w \in \mathscr{W}_k\}$, the dispatch durations of $\mathscr{W}_k$
        Let s_k be a copy of s_0.
        for $w \in \mathscr{W}_k$ in decreasing order **do**
            Update U_{s_k}, solve PCTSP$(d_w \ell, U_{s_k}, \{\beta_i \mathbb{P}(\tau_i \geq w)\})$ and add optimal route to s_k at w.
        LOCALSEARCH(s_k)
        if $(c_{s_k} < c_{s_0})$ **then** $s_0 \leftarrow s_k$
    return s_0

8.2 Proof of Proposition 5.1: Rolling out the *a priori* optimal solution guarantees nonnegative expected cost savings

We base this proof in the result provided by [25] stating that any *sequentially improving* heuristic weekly improves its performance when rolled out. The definition of sequentially improving heuristic is stated in Definition 8.1. We just need to prove that the optimal *a priori* solution is a sequentially improving heuristic. This fact is stated in Proposition 8.2.

Definition 8.1. *(Sequentially Improving Policy). Let s be the state of the system, let $\pi^{H(s)}$ be a policy induced by a Heuristic $H(s)$ computed at state s, and let $C^{\pi}(s)$ be the expected cost-to-go paid if a policy π is implemented from state s onwards. Let s' be any state such that it is on a sample path induced by implementing $\pi^{H(s)}$ at state s. Then, $H(\cdot)$ is sequential improving if*

$$\mathbb{E}\left[C^{\pi^{H(s)}}(s')|s'\right] \geq \mathbb{E}\left[C^{\pi^{H(s')}}(s')|s'\right] \tag{6}$$

Proposition 8.2. *The optimal a priori solution is sequentially improving*

Proof. (Proposition 8.2) Let $\pi^{AP(s)}$ be the resulting policy that arises from the optimal *a priori* policy computed in state s and let s' a state of the system in the sample path induced by $\pi^{AP(s)}$. Assume by contradiction that we have $\mathbb{E}\left[C^{\pi^{AP(s)}}(s')|s'\right] < \mathbb{E}\left[C^{\pi^{AP(s')}}(s')|s'\right]$, then $\pi^{AP(s)}$ has expected cost lower than the expected cost of the optimal *a priori* solution policy computed at state s'. This is clearly a contradiction. $\square$

8.3 Algorithmic details for the greedy *a priori*-based policy

Let $\mathscr{W}^{AP}$ be the set of waves where vehicle dispatches take place in the *a priori* solution, and let $Q_w \subset N_w$ be its set of planned orders to be dispatched at each $w \in \mathscr{W}^{AP}$. We implement the heuristic dynamic policy outlined in Algorithm 9.

Algorithm 9 Greedy *a priori*-based policy

1: Set $w \leftarrow \max\{v \in \mathscr{W}^{AP}\}$, $w^+ \leftarrow \max\{v \in \mathscr{W}^{AP} : v < w\}$
2: Wait at the depot until wave w
3: **while** $w > 0$ **do**
4:     Read system state (w, R, P)
5:     Compute $\bar{R} := R \setminus \bigcup_{v=1}^{w^+} Q_v$, the set of open orders not included in future dispatches of the *a priori* solution
6:     Solve PCTSP$((w - w^+)\ell, \bar{R}, \beta)$ and let $S \subset \bar{R}$ be the optimal set of sequenced orders selected
7:     **if** $q_w(S) = w^+$, **then** dispatch a vehicle to S, set $w \leftarrow w^+$, $w^+ \leftarrow \max\{v \in \mathscr{W}^{AP} : v < w\}$,
8:     **else** set $w \leftarrow w - 1$.

8.4 Set 1: Further empirical analysis for an average solution's dispatch structure

The graphs presented in Figures 10,11, and 12 show for each policy the evolution of the average number of dispatches, duration in waves per dispatch, and initial number of waves spent waiting at the depot over the different instance parameters. From the first graphs on the left, we see that all policies dispatch more vehicles as n grows and wait a smaller amount of initial waves at the depot, meaning that a relatively dense geography justifies a higher number of dispatches during the day and more "active" dispatch waves during the operating period. We also see that the difference in routes dispatched between RP and AP increases with n, and that dynamic policies slightly reduce the waves used per dispatch as n grows. This shows empirically how dynamic policies increase recourse opportunities, *i.e.*, more returns to the depot, as n increases. The second set of graphs from left to right show how our policies have fewer and longer vehicle dispatches with shorter initial waiting periods as off-line information increases (higher p_{start}). This means that as deterministic information increases, fewer recourse opportunities are needed and routing efficiency becomes the focus. The shorter initial wait periods can be explained because there may be relatively less need to "wait and see" versus instances where less information is given at start. Similar effects can also be observed from the graphs on the right. It is interesting to note that RP slightly increases the average number of dispatches and the route length over AP as p_{out} and σ increase. Again, it shows how this policy can recover from uncertainty, *e.g.*, orders realizing later than expected or not showing up, by dynamically inserting unplanned orders as substitutes.

8.5 Set 2: Further empirical analysis of the tradeoff between customer service and routing efficiency under different values the data set parameters p_{start}, p_{out} and σ

We next analyze the performance of all heuristics under different values of α and the data set parameters p_{start}, p_{out} and σ. In Figure 13 we present the previously shown Pareto charts for all heuristic policies under different settings of p_{start}. We observe that RP can approximately cut the gap between the *a priori* heuristic and the perfect information bound in half. Approximately two thirds of that improvement can be achieved by GP. When p_{start} is small (10%) the value of dynamic policies is higher and the marginal rate of substitution

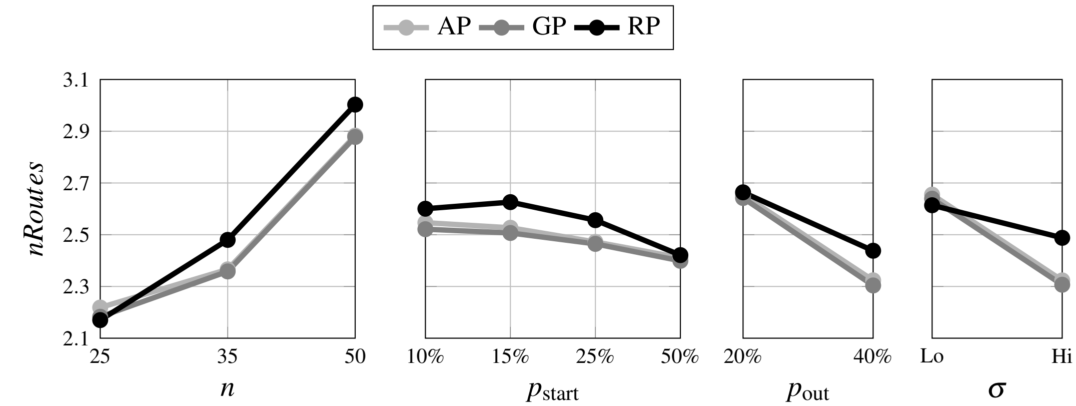


Figure 10: Average number of vehicles dispatched versus n, p_{start}, p_{out}, and σ

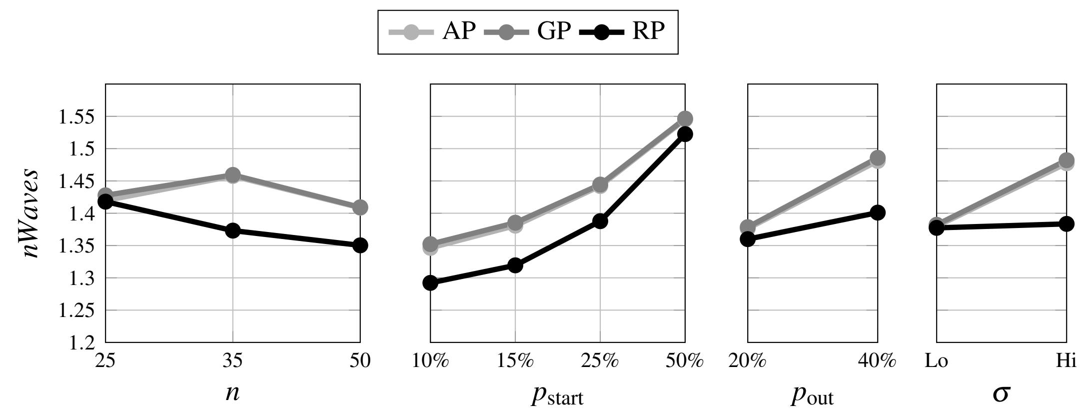


Figure 11: Average number of waves per vehicles dispatch versus n, p_{start}, p_{out}, and σ

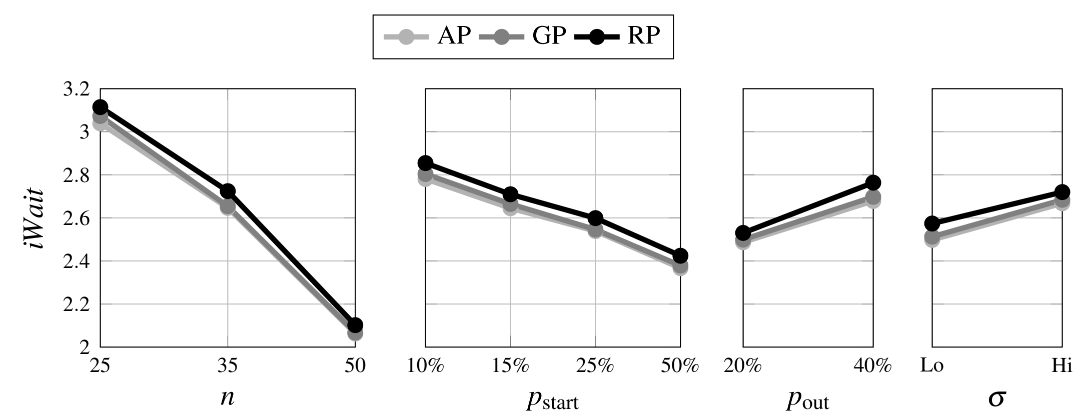


Figure 12: Average number waves waited before initial dispatch versus n, p_{start}, p_{out}, and σ

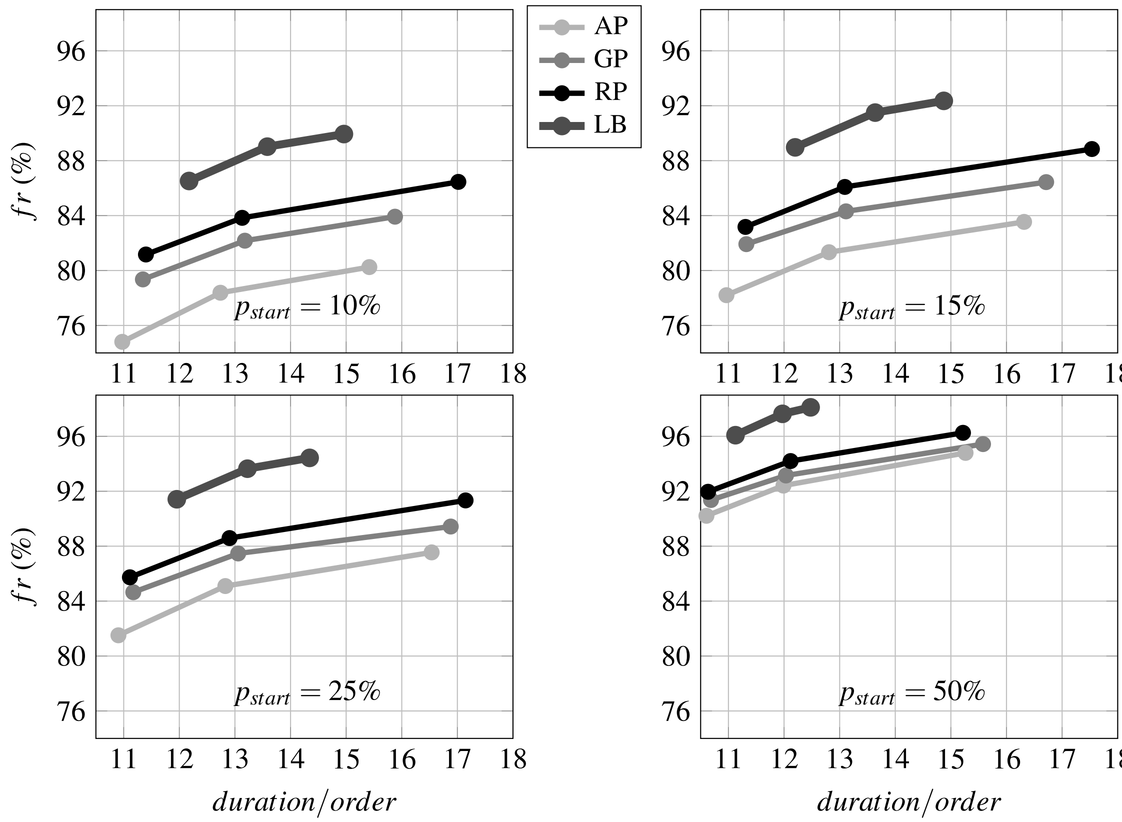


Figure 13: Pareto charts showing each policy's average fr and $duration/order$ for different settings of p_{start}

between fr and $duration/order$ is smaller. When p_{start} in higher (50%) there is less additional value in implementing dynamic policies and the marginal rate of substitution becomes higher, implying that one has to sacrifice less routing efficiency for a marginal increase in order coverage. We also see that as p_{start} grows the system moves to the upper left, better for both fr and $duration/order$.

In Figure 14 we present the Pareto charts of all heuristics for each value of the conditional probability of a order not showing up, p_{out}. We observe that as p_{out} increases all heuristics move to the right, a loss in routing efficiency explained by the increased amount of uncertainty in the order arrival process that makes *a priori* plans less reliable; this is especially true for problems maximizing fill rate, making the marginal rate of substitution smaller. Dynamic policies provide more value as p_{out} increases and the curves move up in relation to AP, because they can better recover from unexpected order late arrivals and corresponding changes in the route plans.

In Figure 15 we analyze the Pareto charts of all heuristics under the two settings of order ready wave variability (σ). We observe two effects as variability increases. First, all curves get wider, with a smaller marginal rate of substitution. This implies that as arrival variability increases it is more expensive to gain coverage and more sacrifices in efficiency have to be made. Also, we observe that AP and GP move down as σ increases, meaning that our static policy and simple dynamic policy lose order coverage. This is not the case of RP; again we see the importance of sophisticated dynamic policies as variability increases.

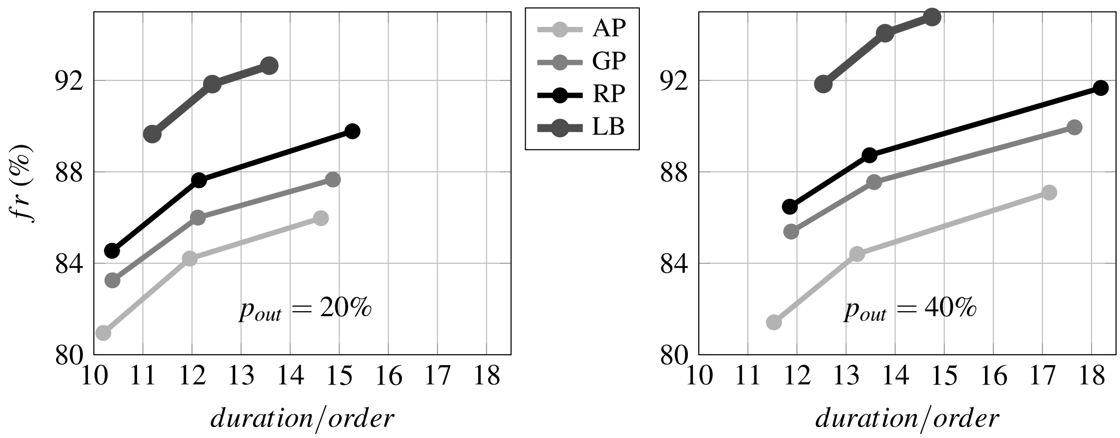


Figure 14: Pareto charts showing each policy's average fr and $duration/order$ for different settings of p_{out}

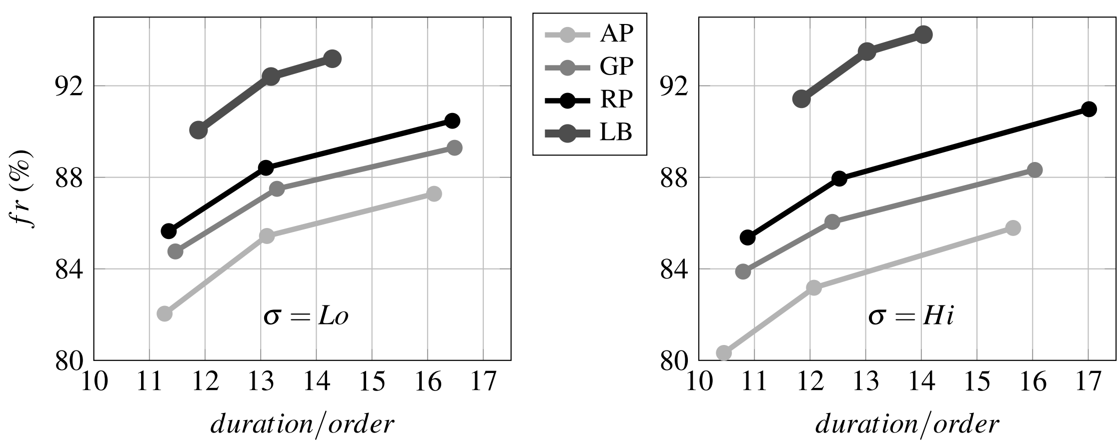


Figure 15: Pareto charts showing each policy's average fr and $duration/order$ for different settings of σ