

# The One-Dimensional Dynamic Dispatch Waves Problem

Mathias Klapp   Alan L. Erera   Alejandro Toriello

H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology

Atlanta, Georgia 30332

maklapp at gatech dot edu, {aerera, atoriello} at isye dot gatech dot edu

June 30, 2015

## Abstract

We study same-day delivery (SDD) systems by formulating the Dynamic Dispatch Waves Problem (DDWP), which models a depot where delivery requests arrive dynamically throughout a service day. At any dispatch epoch (wave), the information available to the decision maker is (1) a set of known, open requests which remain unfulfilled, and (2) a set of potential requests that may arrive later in the service day. At each wave, the decision maker decides whether or not to dispatch a vehicle, and if so, which subset of open requests to serve, with the objective of minimizing expected vehicle operating costs and penalties for unserved requests. We consider the DDWP with a single delivery vehicle and request destinations on a line, where vehicle operating times and costs depend only on the distance between points. We propose an efficient dynamic programming approach for the deterministic variant, and leverage it to design an optimal *a priori* policy with predetermined routes for the stochastic case. We then show that fully dynamic policies may perform arbitrarily better than *a priori* ones, and propose heuristics and dual bounds for this case.

**Keywords:** same-day delivery, dynamic dispatch, approximate dynamic programming

## 1 Introduction

E-commerce and the home delivery channel continue to grow within the consumer retail industry sector. According to Forrester [50], the online sector accounted for 9% of the \$3.2 trillion U.S. retail industry sales in 2013 and is forecast to grow 10% per year through 2018. Sales and density growth can help reduce last-mile logistics costs. However, the online retail segment is extremely competitive and operates with very low margins, driving a need for continued logistics optimization. Consider the case of Amazon. Its 2013 annual report shows an operational margin of just 1% on annual revenue of \$74.5 billion; among expenditures, cost of sales accounts for the largest fraction at 73.5%, but fulfillment costs (11.6%) are also large [31].

Table 1: Examples of same-day delivery pilot programs in the US

Service	What	Charge/order	US cities implemented
Amazon SDD	items from warehouses	\$8.99, \$5.99 members (\$99/year)	ATL,BAL,BOS,CHI,DA-FW,IND,LA,NY,PHI,PHO,SJ-SFO,SEA,DC
Google Express	items from associates	free, but \$10/month (\$15 min order)	SJ/SFO,NY,LA
Instacart	personal shoppers	\$3.99 or \$99/year (\$35 min order)	ATL,AUS,BOS,CHI,DC,DEN,HOU,LA,NY,PHI,SJ-SFO,SEA
Walmart to Go	items from stores	\$10	BAL,DEN,MIN,PHI,SJ-SFO

One enhancement to customer service in this sector is **same-day delivery** (SDD). Several e-retailers and logistics service providers have introduced programs in major US cities; see Table 1. We define SDD as a distribution service where consumers place orders on the same day that they should be delivered. For companies that implement SDD, it is imperative to both offer high levels of customer service and keep logistics costs as low as possible.

Providing SDD services requires two core logistics processes: (1) order management at the stocking location, including receiving, picking, and packing orders; and (2) order distribution from the stocking location to delivery locations. To date, two classes of service providers have deployed SDD services: retailers offering items primarily from owned stocks (in distribution centers or retail stores) such as Amazon and Walmart, and logistics providers serving as intermediaries that pick up packages from stocking locations and deliver them to customers, such as Google and Instacart as well as USPS, FedEx and UPS. Retailers must manage both core logistics processes, while logistics providers are typically concerned with the second one.

This research effort is a study of primary tradeoffs in SDD distribution. We formulate the **Dynamic Dispatch Waves Problem (DDWP)** as a Markov Decision Process (MDP). The DDWP models the dynamics of a single dispatch facility (depot) where customer order requests arrive dynamically throughout an operating day. At any decision epoch, which we call a *wave*, the logistics operator maintains a set of known open requests with known delivery destinations and a set of potential requests that may arrive before the end of the day. At each wave, the operator decides whether or not to dispatch vehicles loaded with known orders, and the vehicle routes for dispatched orders. The objective is to minimize expected operational costs and expected penalties for unserved open requests at the end of the day. Such penalties could represent the cost of direct dispatch or revenue lost due to unserved customers.

Dynamic optimization problems like the DDWP are those where only a subset of all relevant information

is known prior to the initial decision epoch, and the rest is revealed over time during the operating day. An optimal solution to such problems is a *dynamic policy* that determines best decisions given the information state available at each decision epoch. In contrast are simpler *a priori* policies that specify certain decisions in advance, and may allow simple changes via recourse rules. A DDWP instance is characterized by its *degree of dynamism*, a ratio of the amount of information revealed dynamically (online) to that available offline; see [44]. When this ratio is large, dynamic policies may substantially outperform *a priori* policies.

We study the interaction between two important decisions in SDD distribution systems: dynamic dispatch and vehicle routing. Dispatch decisions refer to selection of the times at which vehicles are dispatched and the orders that they deliver, while vehicle routing decisions refer to the sequences of deliveries for each dispatched vehicle. Two fundamental tradeoffs exist. First, there is a tradeoff between waiting and dispatching a vehicle to serve requests. When a vehicle is dispatched, the queue of open requests is reduced but an opportunity to serve future requests during the route is missed. On the contrary, when an available vehicle is not dispatched, we reduce the time remaining in the operating day and potentially increase the likelihood that future requests cannot be served. Second, there is a tradeoff between dispatching longer, time consuming vehicle routes versus shorter ones. On one hand, a route serving many requests uses more total travel time, and therefore keeps the vehicle away from the depot longer, but requires less time per customer visited. On the other hand, a shorter route uses more time per customer, but returns to the depot faster and enables the vehicle to be reused sooner for future orders.

To simplify the vehicle routing decisions, this paper focuses on problem instances where a single vehicle is available to make deliveries to customer locations on the positive real line with the depot as the origin; travel times and vehicle operating costs are proportional to distances between points.

We consider the following to be our main contributions.

1. We formulate the DDWP to capture the basic aspects of dynamic dispatch, order selection, and routing decisions for same-day delivery.
2. We develop an approach for determining optimal *a priori* solutions to the stochastic one-dimensional variant by reducing this problem to an equivalent deterministic problem where all customer request arrival times are known in advance.
3. We show that, although *a priori* policies work well in practice, there exist problem instances with high degree of dynamism for which these solutions are arbitrarily worse than optimal dynamic policies.

Accordingly, we provide two schemes to obtain dynamic policies for the one-dimensional problem. The first is a rollout of the *a priori* policy, and the second is an approximate dynamic programming approach that uses an approximate linear program (ALP) to approximate the cost-to-go function. We empirically show the benefits of dynamic policies with computational experiments over two sets of representative instances.

The remainder of the paper is organized in the following manner. Section 2 contains a literature review, Section 3 formulates the model, 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. An Appendix contains all technical proofs not included in the body of this document.

## 2 Literature Survey

### 2.1 Vehicle Routing and Dispatch Problems

The deterministic Vehicle Routing Problem (VRP) and Traveling Salesman Problem (TSP) have been studied extensively; see *e.g.*, the texts [6, 30] for the TSP and [19, 28] for the VRP. Dynamic and stochastic VRPs are problem extensions where some parameters are unknown during planning and/or operations. The simplest stochastic VRP problems are *a priori* optimization models, where fixed operating (recourse) rules are used to modify the solution during operation; see [14, 19, 26] for recent surveys. Dynamic and online VRPs are problems where information is revealed over time during the operating period, and routing and scheduling decisions are updated in response; see [34, 38, 44, 46, 52]. Different stochastic and dynamic VRPs focus on uncertainty in different sets of parameters. Some examples are the VRP with stochastic demands [2, 11, 29, 43, 49], the VRP with stochastic travel times [18, 35, 36, 39, 40, 41, 51, 53], and the VRP with probabilistic customers [8, 15, 24, 32, 33, 37, 54].

A relevant problem in the literature that incorporates the dispatch dimension is the Dynamic Multiperiod Routing Problem (DMPRP) [4, 5, 55], which consists of a distribution center dispatching orders with a single vehicle and a planning horizon divided into time periods (typically days). Customer orders arrive online at the beginning of each period with a geographical location, and each one has to be served within two periods. The decision maker must choose which orders to postpone and which ones to serve immediately in the vehicle's route defined by a TSP. A recent extension called the Dynamic Multiperiod Vehicle Routing Problem with Probabilistic Information (DVRPP) [3] covers the dynamic-stochastic case, where probabilis-

tic information about future requests arrival times and service time windows is available at each decision epoch. In this case, there is a fleet of capacitated vehicles available at the depot, and each request demands upon arrival a previously known product quantity that should be served within its service time window. The problem is solved heuristically using a prize-collecting VRP model that outputs which orders to serve in each decision epoch. The prizes of open requests are approximated as increasing functions of the proximity to the service deadline and decreasing functions over the geographical closeness to potential future arrivals. Although related to the DDWP, this model does not work for the same-day problem, since it assumes that routing occurs between consecutive periods of potentially unlimited time duration. In the case of SDD, it is fundamental to consider route duration constraints and incorporate how these constraints affect the quantity and duration of future dispatch periods, *e.g.*, shorter routes allow more dispatches per day. Moreover, a model that intends to serve all requests ignores the relative importance between different requests.

The closest dispatch-related problem found in the literature to the DDWP is perhaps the VRP with release dates (VRP-rd) [7, 16] that considers a deterministic problem with a depot dispatching orders with previously known release dates. A release date specifies the earliest time when the order can be picked up at the depot before it is transported to its customer's location. In [7] the authors study simplified versions of the problem, with two variants being particularly relevant to our work: one minimizes the time required to serve all orders by a single vehicle, and another minimizes the total travel distance subject to a time budget. The authors provide polynomial time algorithms for the cases in which requests are placed over the line and other simplified topologies. In [16], the authors study an extension of the VRP-rd in a general network topology that incorporates service time windows, capacity constraints and a homogeneous fleet of capacitated vehicles. The authors provide heuristic solutions based on genetic algorithms combined with local search procedures. Our problem differs from the VRP-rd by its dynamic-stochastic nature and its prize-collecting features. Both elements are fundamental for SDD, where the problem's context is stochastic and where limited time resources do not allow full coverage of requests most of the time. We require a model capable of selecting the least expensive set of orders to be left unattended or to be covered by more expensive transportation modes.

In addition to same-day delivery with simplified routing costs, one-dimensional models have application in other areas where a vehicle or machine's movement is constrained along a single dimension; see [22, 23, 56]. Also, our model is closely related to the Order Batching Problem (OBP) [25, 42], which determines optimal assignment of pick orders to batches and the pick tour sequence for each batch in warehouse

operations. The Dynamic Order Batching Problem (DOBP) is an extension with orders arriving dynamically while the decision maker processes previous orders; see [13]. Recently, [17] present an analytical model to determine the timing and the number of batches in an order fulfillment system. To the best of our knowledge, none of these models allow a detailed selection of the orders within a batch, and solely contemplate threshold rules such as consolidating the batch when a number (to be determined) of orders have arrived.

## 2.2 MDP and Approximate Dynamic Programming

Most dynamic and stochastic VRPs can be modeled as MDPs; see [47]. Exact solutions to MDP models may not be possible due to the curse of dimensionality, through it is often tractable to develop bounds on optimal costs; see [21] for a survey. An example is the *a posteriori* bound [49] or Perfect Information Relaxation (PIR) [12] that disregards the solution’s “non-anticipative” dynamics and finds the deterministic optimal cost for each possible realization of the random parameters before computing the expected cost. Another bounding technique is the Approximate Linear Programming (ALP) method [20, 48] that looks for suboptimal solutions of the MDP’s dual LP formulation. The fundamental idea is to eliminate the exponential number of state variables by enforcing a dependence on a previously determined low-dimensional set of basis functions. Moreover, its solution can be used as a cost-to-go approximation in heuristic policies. The ALP approach has been successfully applied in stochastic routing before, e.g. [1, 53].

In terms of approaches for the VRP and similar dispatch problems, the curse of dimensionality necessitates approximate dynamic programming (ADP) solution techniques, e.g., [27, 29, 43, 49]. One widely used ADP method is to develop approximations for the optimal cost-to-go function and use it to select an approximately optimal action at any encountered state; see [9, 45]. Rollout algorithms [10] have been widely applied for stochastic routing models, e.g., [29, 49, 53].

## 3 The One-Dimensional Dynamic Dispatch Waves Problem

### 3.1 Problem Definition

Consider a dynamic dispatch and routing problem for a single vehicle operating over a fixed-duration operating period (*i.e.*, a day). The vehicle is dispatched from a depot, located at one end of a line segment, to serve a set of customer delivery requests. After completing a route, the vehicle returns to the depot and may be dispatched again until the end of the operating period. At each decision epoch, the vehicle (if available)

may be dispatched to serve any *open* customer requests, those that have arrived and are ready for dispatch. In addition to information about open orders, probabilistic information describing unknown future order requests is also available. The objective is to minimize vehicle operating costs and penalties for unserved requests. We consider a specific class of problems of this type:

1. Let  $\mathcal{T} := \{1, \dots, T\}$  be the set of *waves* (decision epochs) during the operating period, where waves are counted backwards so that the waves number represents the “waves-to-go” before  $t = 0$ , the deadline for the vehicle to return to the depot. Let  $\mathcal{T}_0 = \mathcal{T} \cup \{0\}$ .
2. Let  $N := \{1, \dots, n\}$  be the set of all *potential* customer requests  $i \in N$ , where each  $i$  is characterized by: (1) a known destination represented by a round-trip travel time of  $d_i$  from the depot; (2) a penalty cost  $p_i > 0$  that must be paid if the request arrives but is not served by  $t = 0$ ; and (3) a random arrival time  $\tau_i$  drawn from a request-dependent distribution with support  $\mathcal{T} \cup \{-1\}$ , where  $-1$  indicates “no arrival”. Let  $N$  be ordered such that  $d_i \leq d_j$  for  $i < j$ . Note that our probabilistic model enumerates all possible request arrivals. Another way of modeling this problem is to define a fixed set of locations at which orders appear with potentially multiple arrivals per wave. This alternative probabilistic model is implicitly captured in our setting by adding several requests with an identical customer location.

A vehicle located at the depot at any wave  $t \in \mathcal{T}$  can be dispatched to serve some subset  $S$  of the set of open (revealed and unattended) requests  $R \subseteq \{i \in N : \tau_i \geq t\}$  at wave  $t$ . Once a vehicle leaves the depot at wave  $t$ , it cannot serve any request arriving at  $\tau < t$  until it returns for reloading; we assume that once dispatched, a vehicle must finish its route. Serving request set  $S$  requires time, and we assume that no additional service time is required beyond vehicle travel time. Given request locations along the line, the time required by the vehicle to serve  $S$  is then  $d_S := \max_{i \in S} d_i$ ; we assume vehicle operating cost for this dispatch is  $\alpha d_S$ . A vehicle dispatched at  $t$  returns to the depot at  $t - d_S$ .  $S$  is therefore constrained by  $d_S \leq t$ , but we assume no other constraints on  $S$ , such as capacity, consistent with motivating SDD applications where time is the binding resource. Total system cost is measured by the sum of the vehicle operating costs over all dispatches plus the sum of the penalties  $p_i$  for all  $i \in R$  at the terminal wave ( $t = 0$ ).

For purposes of analysis, we suppose in this paper that time between consecutive waves is constant and equal to the time required for the vehicle to complete a round trip with dispatch travel time 1. Furthermore, suppose that the  $d_i$  values are scaled such that they are all integer.

### 3.2 MDP formulation of the DDWP

We now formulate an MDP for the DDWP. At each wave  $t \in \mathcal{T}_0$ , the system state is given by  $(t, R, P) \in \mathcal{S}$ , where  $\mathcal{S}$  is the state space,  $t$  represents the waves-to-go,  $R$  is the set of open requests, and  $P$  is the set of remaining *potential* requests with an unknown arrival time  $\tau < t$ . Requests not in  $R$  or  $P$  have been already served and so the pair  $(R, P)$  belongs to the set  $\Xi := \{(R, P) : R \cup P \subseteq N, R \cap P = \emptyset\}$ . The maximum number of waves and the three possible states for each requests (open, potential and served) define a bound on the cardinality of the state space given by  $\mathcal{O}(3^n T)$ .

In any non-terminal state  $(t, R, P)$  with  $t \geq 1$ , we choose between waiting with the vehicle at the depot, and dispatching the vehicle to serve a set of requests  $S \subseteq R$ , which is equivalent to selecting a route of duration  $d \leq t$  serving the set  $\{i \in R : d_i \leq d\}$ . Define then the action space  $\mathcal{A}_R^t := \{d_i \mid \forall i \in R : d_i \leq t\}$ , with cardinality  $\mathcal{O}(n)$ . Selecting an action  $d$  in a given state  $(t, R, P)$  transforms the state as follows. If a dispatch of length  $d$  is selected,  $R$  is partitioned into the new set of unattended requests  $R_d := \{i \in R : d_i > d\}$  and the set of served requests  $\bar{R}_d = R \setminus R_d$ . Time moves forward to  $t - d$  and state  $(t, R, P)$  becomes  $(t - d, R_d \cup F_d^t, P \setminus F_d^t)$  where  $F_d^t := \{i \in N : t > \tau_i \geq t - d\}$  is the set of newly arriving requests. If no dispatch occurs ( $d = 0$ ), the new state is  $(t - 1, R \cup F_1^t, P \setminus F_1^t)$ .

Let  $C_t(R, P)$  be a set function representing the minimum expected cost-to-go at state  $(t, R, P) \in \mathcal{S}$ . The optimal expected cost  $C^*$  is defined recursively over  $t \in \mathcal{T}_0$  in (1), where  $\hat{R}$  is the set of known open requests at the start of the horizon ( $t = T$ ). First, at  $t = 0$  the cost-to-go is simply the sum of penalties of unserved requests, and subsequently, for each  $t \in \mathcal{T}$  the cost-to-go at state  $(t, R, P)$  is equal to the minimum cost between no dispatch and a dispatch to any distance  $d \in \mathcal{A}_R^t$ :

$$C_0(R, P) = \sum_{i \in R} p_i \quad \forall (R, P) \in \Xi \quad (1a)$$

$$C_t(R, P) = \min_{d \in \mathcal{A}_R^t \cup \{0\}} \left\{ \alpha d + \mathbb{E}_{F_d^t} [C_{t - \max\{1, d\}}(R_d \cup F_d^t, P \setminus F_d^t)] \right\}, \quad \forall t \in \mathcal{T}, (R, P) \in \Xi \quad (1b)$$

$$C^* = \mathbb{E}_{\hat{R}} [C_T(\hat{R}, N \setminus \hat{R})]. \quad (1c)$$

This formulation considers an uncertain set  $\hat{R}$ , but a useful special case is when  $\hat{R}$  is known. The optimal action  $d_t^*(R, P) \in \mathcal{A}_R^t \cup \{0\}$  that attains  $C_t(R, P)$  is then defined as a set function for each state  $(t, R, P)$ . The vector of optimal actions for each state is called an optimal policy. We can also express optimality conditions



using a standard LP dual reformulation of (1),

$$C^* = \max_{\{C \geq 0\}} \mathbb{E}_{\hat{R}} [C_T (\hat{R}, N \setminus \hat{R})] \quad (2a)$$

$$\text{s.t. } C_0(R, P) \leq \sum_{i \in R} p_i, \quad \forall (R, P) \in \Xi \quad (2b)$$

$$C_t(R, P) \leq \mathbb{E}_{F_t^t} [C_{t-1} (R \cup F_1^t, P \setminus F_1^t)], \quad \forall t \in \mathcal{T}, (R, P) \in \Xi \quad (2c)$$

$$C_t(R, P) \leq \alpha d + \mathbb{E}_{F_d^t} [C_{t-d} (R_d \cup F_d^t, P \setminus F_d^t)], \quad \forall t \in \mathcal{T}, (R, P) \in \Xi, d \in \mathcal{A}_R^t, \quad (2d)$$

which very clearly shows the difficulty in finding an optimal policy; formulation (2) has exponentially many variables, exponentially many constraints and exponentially many terms in the expectations.

## 4 A Priori Solutions for the Stochastic DDWP

In this section we develop *a priori* policies for the DDWP defined in (1). We begin studying the deterministic version of the problem to understand the structure of optimal *a priori* policies.

### 4.1 The Deterministic Case

Suppose arrivals are known with certainty at the beginning of the horizon, and let the set of arriving requests be  $N_A := \{i \in N : \tau_i > 0\}$ . Requests still arrive dynamically over the operating period, and thus it remains infeasible to serve a request with a vehicle dispatch prior to its arrival time.

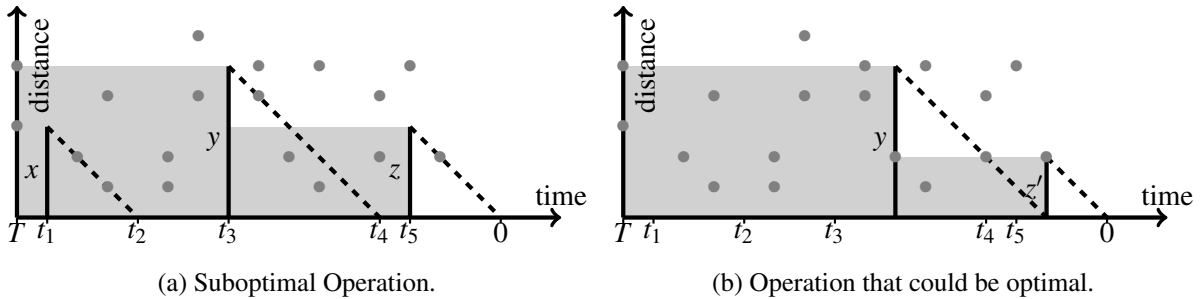


Figure 1: Examples of vehicle operations described in the distance versus time graph.

Figure 1a gives an instance where arrival times and destinations for each request  $i \in N_A$  are represented

by a coordinate  $(\tau_i, d_i)$  in a distance versus time graph. Also depicted is an example vehicle dispatch plan. The vehicle starts at the depot at wave  $T$  and waits until  $t_1$  when it is dispatched a distance  $x$ . Then, it returns at  $t_2 = t_1 - x$  and waits until  $t_3$  to execute a second dispatch of distance  $y$ , and so on. Requests covered by this operation are those with coordinates inside the shaded areas. We now state and prove three properties that at least one optimal vehicle dispatch plan should satisfy:

**Property 4.1** (Decreasing consecutive dispatches). *For all dispatch pairs starting at two waves  $t > t'$  with respective dispatch durations  $d$  and  $d'$ , we have  $d > d'$ .*

*Proof.* If  $d' \geq d$ , by deleting the dispatch at  $t$  we reduce operational cost with unaltered coverage.  $\square$

**Property 4.2** (No wait after a dispatch). *The vehicle does not wait once the first dispatch has occurred.*

*Proof.* If a solution waits for  $w$  waves after a dispatch at  $t$ , we can shift forward each vehicle dispatch that occurs prior to wave  $t$  exactly  $w$  waves in time without reducing the set of covered requests.  $\square$

**Property 4.3** (Dispatch duration equals round-trip time to some request). *The duration of each dispatched route equals  $d_S = \max_{i \in S} d_i$ , where  $S \subseteq R$  is the set of requests served by the route.*

*Proof.* If  $d > d_S$ , by setting  $d = d_S$  we reduce operational cost with unaltered request coverage.  $\square$

Figure 1b depicts an operation that satisfies all properties. A direct consequence of these properties is that we can formulate a deterministic dynamic program with a reduced state space.

Let the set of possible dispatch durations be  $\mathcal{D} := \{d_i \mid \forall i \in N_A : d_i \leq \tau_i\}$ . We can find an optimal dispatch plan via a dynamic program with states  $(t, x)$ , where  $t$  is the current wave and  $x$  is the duration of the previous dispatch completed at wave  $t$  ( $x = 0$  if no dispatches have occurred prior to  $t$ ). Figure 2 is an example of the system at state  $(t, x)$ , where the last dispatch was of duration  $x$  at wave  $t + x$  and covered all requests shaded in light gray. Requests shaded in medium gray will never be served by an optimal dispatch plan satisfying the previous three properties and are thus lost, and the requests shaded in dark gray could be covered by the next dispatch at wave  $t$ .

An action given state  $(t, x)$  is defined as the next dispatch duration  $d \in \mathcal{A}_{t,x}$ , where

$$\mathcal{A}_{t,x} = \{d_i \mid \forall i \in N_A : d_i \leq t, d_i < x, t \leq \tau_i < t + x\}, \quad \forall x \in \mathcal{D} : t + x \leq T \quad (3a)$$

$$\mathcal{A}_{t,0} = \{0\} \cup \{d_i \mid \forall i \in N_A : d_i \leq t, \tau_i \geq t\}. \quad (3b)$$

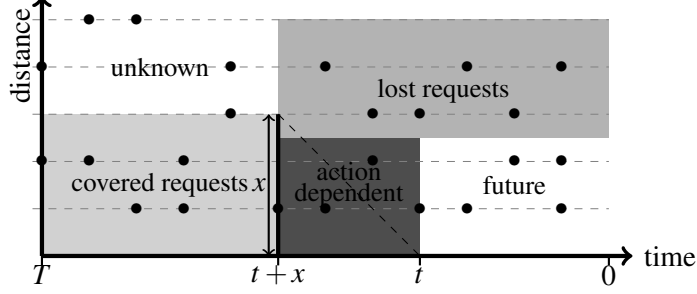


Figure 2: Example of state and action for the deterministic DDWP.

If no dispatches have occurred by  $t$  ( $x = 0$ ), an optimal vehicle operation may wait until  $t - 1$ , *i.e.*,  $d = 0$ .

Define  $C_t(x)$  as the cost-to-go function in state  $(t, x)$ . Optimality equations (solvable in  $\mathcal{O}(n^2T)$  operations) are given by (4), where  $C_T(0)$  is the minimum cost for the deterministic DDWP:

$$C_0(x) = \sum_{i \in N_A} p_i, \quad \forall x \in \mathcal{D} \cup \{0\} \quad (4a)$$

$$C_t(0) = \min_{d \in \mathcal{A}_{t,0}} \left\{ \alpha d - \sum_{i \in N_A: d_i \leq d, \tau_i \geq t} p_i + C_{t-\max\{1,d\}}(d) \right\}, \quad \forall t \in \mathcal{T} \quad (4b)$$

$$C_t(x) = \min_{d \in \mathcal{A}_{t,x}} \left\{ \alpha d - \sum_{i \in N_A: d_i \leq d, t \leq \tau_i < t+x} p_i + C_{t-d}(d) \right\}, \quad \forall t \in \mathcal{T}, x \in \mathcal{D} : t+x \leq T. \quad (4c)$$

In this dynamic program, we initialize by assuming that all arrived requests are not served by  $t = 0$ . When we execute a dispatch of duration  $d$ , we incur its operating cost while also saving the penalties of the requests served.

## 4.2 The stochastic case and *a priori* policies

Consider again the stochastic DDWP defined in (1). We next develop the optimal static *a priori* solution in which a schedule specifying the waves at which to dispatch the vehicle and the duration of each dispatch is determined only with information revealed at the start of the horizon in wave  $T$ .

The operating cost of such an *a priori* solution is known, and the penalties paid for unserved requests depend on the future arrivals. This observation motivates an approach for determining an *a priori* solution that minimizes expected cost. This problem is equivalent to solving a deterministic DDWP instance in which each potential request  $i \in N$  is copied  $T$  times and assumed to arrive at *every* wave  $t \in \mathcal{T}$  for which

its probability of arrival is positive, with an adjusted penalty for not serving the request at wave  $t$  equal to  $p_i P(\tau_i = t | \tau_i < T)$ . Known requests ( $\tau_i \geq T$ ) are not copied and arrive only at wave  $T$  with probability one. Thus, the recursive equations to find an optimal *a priori* policy are a natural extension of the deterministic system (4),

$$C_0^{AP}(x) = \sum_{i \in N: \tau_i = T} p_i + \sum_{i \in N: \tau_i < T} \mathbb{P}(\tau_i > 0 | \tau_i < T) p_i, \quad \forall x \in \mathcal{D} \cup \{0\} \quad (5a)$$

$$C_t^{AP}(0) = \min_{d \in \mathcal{A}_{t,0}} \left\{ \alpha d - \sum_{\substack{i \in N: \\ \tau_i = T, d_i \leq d}} p_i - \sum_{\substack{i \in N: \\ \tau_i < T, d_i \leq d}} \mathbb{P}(\tau_i \geq t | \tau_i < T) p_i + C_{t - \max\{1, d\}}^{AP}(d) \right\}, \quad \forall t \in \mathcal{T} \quad (5b)$$

$$C_t^{AP}(x) = \min_{d \in \mathcal{A}_{t,x}} \left\{ \alpha d - \sum_{\substack{i \in N: \\ \tau_i < T, d_i \leq d}} \mathbb{P}(t \leq \tau_i < t + x | \tau_i < T) p_i + C_{t-d}^{AP}(d) \right\}, \quad \forall t \in \mathcal{T}, x \in \mathcal{D} : t + x \leq T, \quad (5c)$$

where the optimal expected cost is given by  $C_T^{AP}(0)$ . Note that this policy is found by solving a deterministic DDWP, and so, it satisfies Properties (4.1), (4.2) and (4.3).

To estimate the expected cost of implementing this heuristic policy we must see the orders arrived and waiting for service at  $t = T$ . To do this, we assume to know all orders  $i \in N$  with  $\tau_i(m) = T$  for a given set of  $M$  realizations  $m \in \{1, \dots, M\}$  obtained via Monte Carlo sampling. For consistency in computational results, we will use the same  $M$  realizations when comparing performance of lower bounds and different solutions for a given instance.

We can improve the performance of an *a priori* policy by allowing simple recourse actions during the operation. Let a policy be represented by the ordered set of  $k$  dispatches, each with dispatch distance  $d^j$  and wave  $t^j$ :  $\{(d^j, t^j)\}_{j=1}^k$ . The policy satisfies  $t^j - t^{j+1} = d^j$  and  $d^j > d^{j+1}$  for  $j = 1, \dots, k-1$ . Consider the following recourse actions:

1. *Postponement and cancellation*: Consider dispatch  $j$  scheduled at wave  $t^j$ . Any open request  $i$  with  $d_i \leq d^{j+1}$  can be covered by dispatch  $j+1$ . So, if no request  $i$  has arrived since the previous dispatch time  $t^{j-1}$  with  $d^{j+1} < d_i \leq d^j$ , then postpone dispatch  $j$  by modifying its scheduled time  $t^j \leftarrow t^j - 1$  and its duration  $d^j \leftarrow d^j - 1$  if  $d^j - 1 > d^{j+1}$ , otherwise cancel dispatch  $j$ . A rescheduled dispatch is considered again for postponement and cancellation iteratively.

2. *Marginal profit adjustment*: Given a dispatch  $j$  that has not been postponed or cancelled, adjust its

distance  $d^j$  to maximize its marginal profit: Choose actual dispatch distance  $d$  equal to the location  $d_r$  of an open request  $r \in N$  with  $d^{j+1} < d_r \leq d^j$ , such that it maximizes the following marginal profit  $V_r^{mg} := \left( \sum_{\{i \in N: d^{j+1} < d_i \leq d_r, \tau_i \geq t\}} P_i - d_r \right)$ . If  $V_r^{mg} \leq 0$  for each possible request  $r$ , then again postpone the dispatch to  $t^j \leftarrow t^j - 1$  with new dispatch duration  $d^j - 1$  if  $d^j - 1 > d^{j+1}$ , otherwise cancel dispatch  $j$ . If the dispatch is adjusted such that  $d < d^j$ , we also postpone it to time  $t^{j+1} + d$  to potentially serve more customers with no increase in cost.

## 5 Dynamic Policies for the Stochastic DDWP

*A priori* policies, particularly when adjusted via recourse actions, may yield reasonable solutions to many problems. However, there exist instances for which an optimal adjusted *a priori* policy is arbitrarily worse than an optimal dynamic policy.

**Pathological *A Priori* Instances** Consider a family of instances with 2 requests,  $T = 4$ , and a parameter  $z \geq 0$ . Let locations be  $d_1 = 1$  and  $d_2 = 2$ , and penalties  $p_1 = z + 1$  and  $p_2 = z^2 + z + 2$ . Request 1 arrives at  $\tau_1 = 1$ , while request 2 arrives at  $\tau_2 = 3$  with probability  $u = \frac{z}{z+1}$  and  $\tau_2 = 2$  with probability  $v = \frac{1}{z+1}$ . There are four possible *a priori* solutions (See Figure 3). Either of the last two options (c) or (d) are optimal *a priori* policies with simple recourse, and both have expected cost of  $3 + z$ .

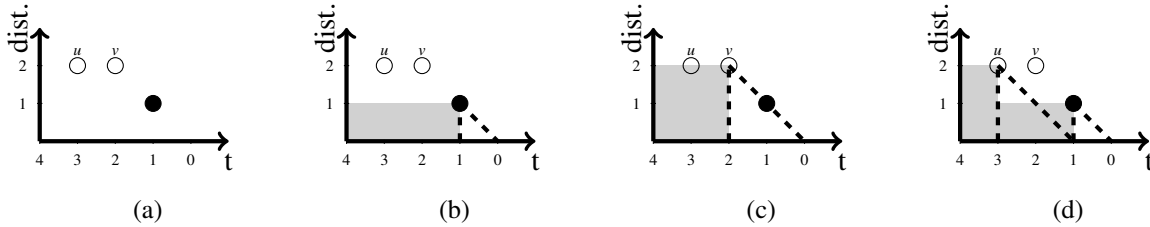


Figure 3: Feasible *a priori* dispatch options.

The optimal dynamic policy is different. If request 2 arrives at  $t = 3$ , it dispatches to  $d = 2$  at  $t = 3$  and then  $d = 1$  at  $t = 1$  for total cost of 3; otherwise it dispatches to  $d = 2$  at  $t = 2$  for cost of  $3 + z$ . The expected cost of this policy is  $3u + (3 + z)v = 3 + \frac{z}{z+1} < 4$ . As  $z \rightarrow \infty$ , the optimal cost is bounded, while any *a priori* policy's cost is unbounded.

## 5.1 A Priori-Based Rollout Policy

One approach to build a dynamic policy is to roll out the *a priori* policy. At each wave  $t \in \mathcal{T}$  when the vehicle is available, we recompute an optimal *a priori* policy given updated information regarding requests (open, potential, and served); if the policy dictates a dispatch  $d > 0$  at  $t$ , the decision is executed and a new *a priori* policy is then computed at  $t - d$ , otherwise a new *a priori* policy is computed at  $t - 1$ . Computing such a rollout policy requires  $\mathcal{O}(n^2T^3)$  operations.

## 5.2 Approximate Linear Programming for the DDWP

Heuristic dynamic policies can be generated via the dual MDP reformulation (2). Because this formulation has exponentially many variables and constraints, the ALP approach restricts its feasible region in such a way that the resulting optimization model is tractable and so it yields a lower bound for the optimal expected cost-to-go that can be used within a rollout policy.

We can generate a lower bound  $C_t^{ALP}(R, P)$  for the cost-to-go function of the DDWP at any feasible state  $(t, R, P)$  by representing  $C_t(R, P)$  as a linear function of a predetermined set of basis functions, and then solving the resulting restriction of (2). Let  $C_t(R, P) \approx C_t^{ALP}(R, P)$ , where

$$C_t^{ALP}(R, P) := \sum_{i \in R} a_i^t + \sum_{i \in P} b_i^t - \sum_{k=1}^t v_k, \quad (6)$$

and where  $a_i^t$  represents the cost of request  $i$  if it is open at wave  $t$ ,  $b_i^t$  represents the cost of potential request  $i$  if it hasn't arrived by wave  $t$ , and  $v_k$  represents the incremental value of each wave  $k$ .

**Proposition 5.1.** *Applying the restriction (6) to (2) yields a model equivalent to*

$$C^{ALP} = \max_{\{\mathbf{a}, \mathbf{b}, \mathbf{v}, \mathbf{s}, \mathbf{u}\}} \sum_{i \in N} (\mathbb{P}(\tau_i = T) a_i^T + \mathbb{P}(\tau_i < T) b_i^T) - \sum_{t=1}^T v_t \quad (7a)$$

$$s.t. \ a_i^0 = p_i, b_i^0 = 0, \quad \forall i \in N \quad (7b)$$

$$s_{it} \geq a_i^t - a_i^{t-1}, \quad \forall i \in N, t \in \mathcal{T} \quad (7c)$$

$$s_{it} \geq b_i^t - f_{it} a_i^{t-1} - \tilde{f}_{it} b_i^{t-1}, \quad \forall i \in N, t \in \mathcal{T} \quad (7d)$$

$$\sum_{i \in N} s_{it} \leq v_t, \quad \forall t \in \mathcal{T} \quad (7e)$$

$$u_{it}^d \geq a_i^t - \mathbb{1}_{(d_i > d)} a_i^{t-d}, \quad \forall i \in N, t \in \mathcal{T}, d \in \mathcal{A}_N^t \quad (7f)$$

$$u_{it}^d \geq b_i^t - g_{it}^d a_i^{t-d} - \bar{g}_{it}^d b_i^{t-d}, \quad \forall i \in N, t \in \mathcal{T}, d \in \mathcal{A}_N^t \quad (7g)$$

$$\sum_{i \in N} u_{it}^d \leq \sum_{k=t-d+1}^t v_k + \alpha d \quad \forall t \in \mathcal{T}, d \in \mathcal{A}_N^t \quad (7h)$$

$$\mathbf{u}, \mathbf{s} \geq 0, \quad (7i)$$

where  $f_{it} := \mathbb{P}(\tau_i = t - 1 \mid \tau_i < t)$  is the conditional probability that potential request  $i$  at wave  $t$  arrives at the next wave,  $g_{it}^d := \mathbb{P}(\tau_i \geq t - d \mid \tau_i < t)$  is the conditional probability that potential request  $i$  at wave  $t$  arrives in one of the next  $d$  waves, and also  $\bar{g}_{it}^d := 1 - g_{it}^d$  and  $\bar{f}_{it} := 1 - f_{it}$ .

Any set of values  $\{\mathbf{a}, \mathbf{b}, \mathbf{v}\}$  used to compute  $C_i^{\text{ALP}}(\mathbf{R}, \mathbf{P})$  which are feasible for (7b)-(7i) yield a lower bound of the cost-to-go function at any state  $(t, \mathbf{R}, \mathbf{P})$ :  $C_i^{\text{ALP}}(\mathbf{R}, \mathbf{P}) \leq C_i(\mathbf{R}, \mathbf{P})$ . In particular, we have  $C^{\text{ALP}} \leq C^*$ .

The proposition's proof is in the appendix. Model (7) has interesting properties which give economic intuition and accelerate computation times; each of these properties is proved in the appendix.

**Property 5.2** (Bounds). *We may assume  $0 \leq a_i^t \leq p_i$  and  $0 \leq b_i^t \leq g_{it}^t p_i, \forall i \in N, t \in \mathcal{T}_0$  without loss of optimality.*

Intuitively, Property 5.2 implies that the individual cost per open request at any wave is nonnegative and cannot exceed the penalty for leaving the request unattended, and that the individual cost for any potential request at any wave is nonnegative and cannot exceed the penalty discounted by the arrival probability.

**Property 5.3** (Lost requests). *Without loss of optimality, we may assume that  $a_i^t = p_i$  for any  $i \in N, t \in \mathcal{T}_0 : d_i > t$ , and  $b_i^t = g_{it}^t p_i$  for any  $i \in N, t \in \mathcal{T}_0 : d_i \geq t$ .*

Property 5.3 says that the cost of having an open request  $i$  at time  $t$  with an impossible dispatch ( $d_i > t$ ) is equal to  $p_i$ . A similar idea motivates the expression for  $b_i^t$ .

The following theorem, also proved in the appendix, describes the performance of the ALP lower bound in the deterministic case.

**Theorem 5.4** (Strong duality for the deterministic case). *Assume request  $i$ 's arrival wave  $\tau_i$  is deterministic for each request  $i \in N$ . Then the bound given by (7) is tight, i.e., equal to the optimal cost of the deterministic DDWP given in (4).*

The result gives further motivation to use ALP for the DDWP, since the approximation is able to recover optimality in the deterministic case. Furthermore, it relates the ALP and the *a priori* solution: if we transform a stochastic instance into a deterministic one as described in Section 4, the ALP matches the *a priori* solution, and both can be used heuristically. However, the ALP can also be used without the transformation, so it can be viewed as a generalization of the *a priori* rollout policy.

We next apply (7) to approximate the optimal action  $d_i^*(R, P)$ . Given a feasible  $(\mathbf{a}, \mathbf{b}, \mathbf{v})$  to (7b)-(7i), we have a closed linear form lower bound for the expected cost-to-go function measured after a decision with dispatch distance  $d \in \mathcal{A}_R^t$  has been taken via

$$\begin{aligned} \mathbb{E}_{F_d^t} [C_{t-d}(R_d \cup F_d^t, P \setminus F_d^t)] &\geq \mathbb{E}_{F_d^t} [C_{t-d}^{ALP}(R_d \cup F_d^t, P \setminus F_d^t)] \\ &= \mathbb{E}_{F_d^t} \left[ \sum_{i \in R_d \cup F_d^t} a_i^{t-d} + \sum_{i \in P \setminus F_d^t} b_i^{t-d} - \sum_{k=1}^{t-d} v_k \right] \\ &= \sum_{i \in R_d} a_i^{t-d} + \sum_{i \in P} (g_{it}^d a_i^{t-d} + \bar{g}_{it}^d b_i^{t-d}) - \sum_{k=1}^{t-d} v_k. \end{aligned} \quad (8)$$

A similar expression can be obtained to underestimate the expected cost-to-go measured after the vehicle waits for one wave at the depot. We use these bounds to compute an approximately optimal action  $d_i^{ALP}(R, P)$ .

Any feasible set of values  $\{\mathbf{a}, \mathbf{b}, \mathbf{v}\}$  provides an underestimate of the expected cost-to-go in (8). In particular, the tightest lower bound is achieved when maximizing (8) subject to (7b)-(7i). This is a *post state and decision re-optimization* of the ALP in which the values of  $\{\mathbf{a}, \mathbf{b}, \mathbf{v}\}$  are recomputed at each wave  $t$  when the vehicle is ready at the depot, and for each potential action  $d \in \mathcal{A}_{R^t(m)}^t \cup \{0\}$ . We compute the approximate optimal action by

$$d_i^{ALP}(R, P) = \underset{d \in \mathcal{A}_R^t \cup \{0\}}{\operatorname{argmin}} \begin{cases} \max_{\{(\mathbf{a}, \mathbf{b}, \mathbf{v}) \in (7b)-(7i)\}} \sum_{i \in R} a_i^{t-1} + \sum_{i \in P} (f_{it} a_i^{t-1} + \bar{f}_{it} b_i^{t-1}) - \sum_{k=1}^{t-1} v_k, & \text{if } d = 0 \\ \alpha d + \max_{\{(\mathbf{a}, \mathbf{b}, \mathbf{v}) \in (7b)-(7i)\}} \sum_{i \in R_d} a_i^{t-d} + \sum_{i \in P} (g_{it}^d a_i^{t-d} + \bar{g}_{it}^d b_i^{t-d}) - \sum_{k=1}^{t-d} v_k, & \text{if } d \in \mathcal{A}_R^t \end{cases}. \quad (9)$$

This involves solving  $\mathcal{O}(nT)$  linear programs sharing the same feasible set of solutions. Its performance can be improved by applying LP warmstart and ruling out suboptimal dispatch distances (see the appendix for details). The procedure is repeated for each realization  $m = 1, \dots, M$  of arrivals to estimate its expected cost.



## 6 Computational Experiments

We present two sets of computational experiments using different families of randomly generated instances. Our goal is to test the quality of the various heuristics and to obtain qualitative insights regarding solutions. The two sets of experiments differ in their models of the request arrival process. In the first set, we assume that the conditional likelihood of a request arrival by the next dispatch at wave  $t$  is constant over time but may vary by request. In the second set, we use an arrival distribution that assigns probabilities for the arrival time (or the non-arrival event) for each request using a mean arrival that varies by request. All heuristics were programmed in Java and computed using a 2.1GHz Intel Core i7-3612QM processor with 8 GB RAM, using CPLEX 12.4 when necessary as the LP solver.

Table 2 summarizes the lower bounds and heuristic policies’ costs that we computed for the instances in this study. We do not include the ALP lower bound, as our preliminary experiments revealed it to be weaker than the PIR bound. Similar behavior has been observed in other stochastic routing contexts, *e.g.*, [53].

For each particular instance, we simulated  $M = 100$  realizations of the arrival time vector  $\tau$ , and use this common set to estimate lower bounds and policies’ expected costs via Monte Carlo sampling.

Table 2: Lower bounds and heuristic policies’ costs computed

Type	Procedures
Lower bound	perfect information relaxation (PIR)
<i>A priori</i> policies	Static <i>a priori</i> policy (AP) & <i>a priori</i> policy with recourse actions (APR)
Dynamic policies	dynamic <i>a priori</i> policy rollout (DAP) & dynamic ALP policy (DALP)

### 6.1 Design of Instance Set 1: Stationary Conditional Arrival Probability

The first set of experimental instances model arrivals using a stationary conditional arrival distribution for each request. Therefore, for each  $i \in N$ , the probability that it arrives at wave  $t$  given that it has not yet arrived is independent of  $t$ , *i.e.*,  $\mathbb{P}(\tau_i = T) = \mathbb{P}(\tau_i \geq t - 1 \mid \tau_i < t) = \theta_i$  and  $\mathbb{P}(\tau_i = -1) = (1 - \theta_i)^{T-1}$ .

We construct instances with different size, geography, and time flexibility as follows. Let  $(n, \ell, r)$  define an instance where  $n$  is the number of potential requests over the horizon;  $\ell$  is the maximum distance between a request and the depot; and  $r := T/\ell$  is the ratio between the total number of waves  $T$  and  $\ell$ . We consider all combinations of  $n \in \{5, 10, 20, 40, 60, 80, 100\}$ ,  $\ell \in \{5, 10, 20\}$ , and  $r = \{1, 2, 3\}$  and generate 20 random instances for each combination by varying the vectors  $\{\theta_i\}$ ,  $\{p_i\}$ , and  $\{d_i\}$  as

$\{\theta_i\}$ : probability parameter  $\theta_i$  for each  $i$  drawn i.i.d. from a continuous uniform distribution,  $U(\frac{1}{2T}, \frac{2}{T})$ ;

$\{p_i\}$ : penalty parameter  $p_i$  drawn with equal probability from the values  $\{0.25\ell, 0.5\ell, 0.75\ell, \ell\}$

$\{d_i\}$ : distance parameter  $d_i$  drawn with equal probability from the values  $\{1, \dots, \ell\}$ .

## 6.2 Results for Instance Set 1

Figure 4 reports the average duality gap between the PIR bound and the optimal expected cost for small instances ( $n \in \{5, 10\}$ ) where the fully dynamic-stochastic problem is solvable to optimality.

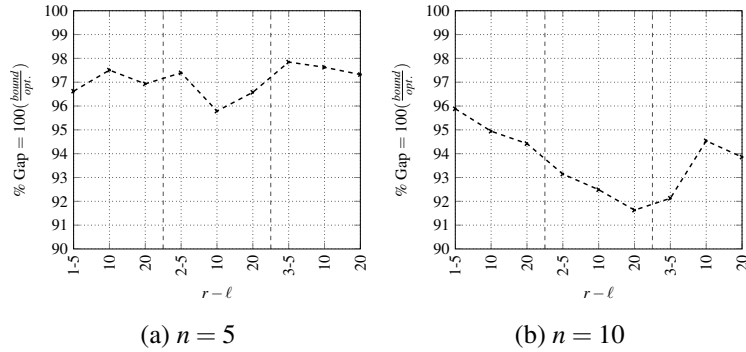


Figure 4: Percentage gap between PIR lower bound and optimal solution values for Instance Set 1

Table 3: Overall performance of heuristics in Instance Set 1

Upper Bound	% GAP vs OPT (small instances)	% GAP vs lower bound	Time per sample-instance (secs)
AP	11.53%	12.14%	0.0124
APR	5.27%	9.24%	0.0122
DAP	1.97%	6.59%	0.1489
DALP	1.65%	6.42%	0.5869

Table 3 presents average gap and solution times for each heuristic. In case of the ALP-based policy (DALP), we employed a hybrid approach that executes DAP until the operation reaches wave  $x\ell$  and, afterwards, executes an ALP-based policy. The motivation is the two policies' complementary behavior. The ALP tends to be too conservative initially, when the remaining horizon includes many possibilities it has to under-approximate, while DAP simply assumes “averages” for the future; conversely, towards the end of the horizon the ALP can more accurately assess possible future recourse actions, and thus can make better decisions. Also, the linear programs in the ALP tend to have highly degenerate polytopes for instances with high flexibility, making them difficult to solve. After searching over a grid of different values in preliminary experiments, we concluded that  $x = 1.1$  yields the best gap while still keeping computation times low. This

contrasts with naive implementations of ALP policies, which can be computationally demanding.

For small instances with  $n = 5$  or  $n = 10$ , Figure 5 shows the average relative gap to the optimal solution. The dynamic *a priori* policy rollout (DAP) and the dynamic ALP-based policy (DALP) dominate the *a priori* solutions and achieve an average gap of 1.97% and 1.65%, respectively.

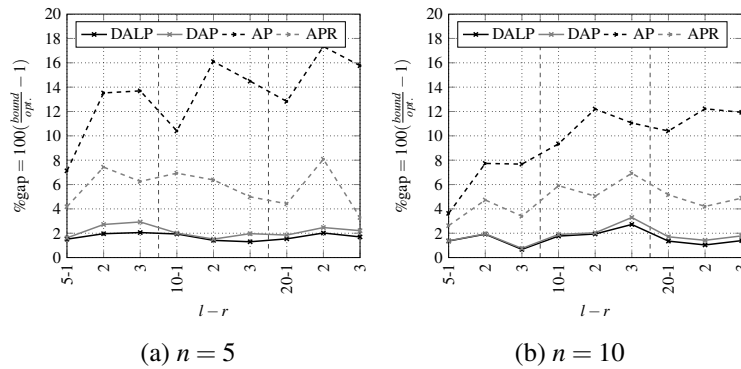


Figure 5: Average percentage gap between heuristic solution costs and optimal costs for Instance Set 1

For larger instances, the gap is computed with respect to the PIR bound. Figure 6 details average gaps over all classes of instances.

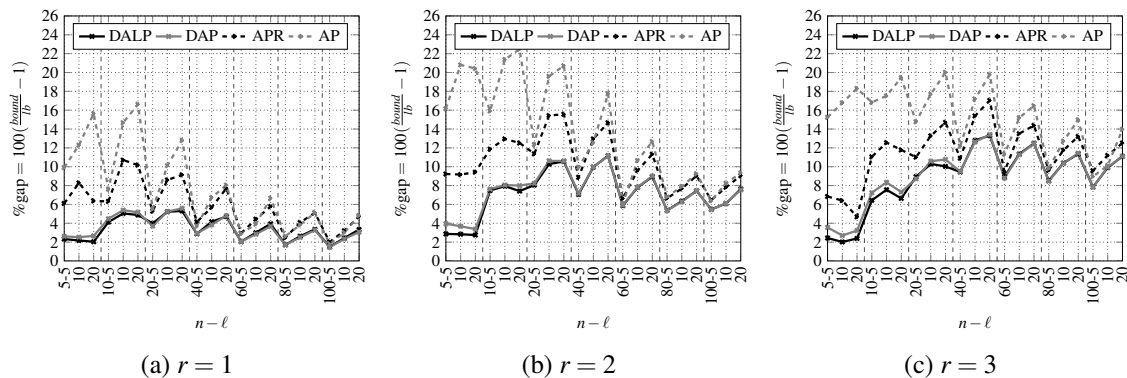


Figure 6: Average percentage gap between heuristic solution costs and lower bound for Instance Set 1

As expected based on each heuristic's recourse possibilities, APR outperforms AP and both are outperformed by the two dynamic policies (DAP and DALP). Also, the gap differences between AP, APR and the dynamic policies decrease with  $n$ . This suggests that dynamic solutions produce a bigger gap improvement for instances with more request arrival granularity, *i.e.*, where an early or late arrival can significantly impact costs unless corrective actions are taken. Conversely, for instances with more requests the marginal

value of dynamic solutions is smaller. This may be due to risk pooling effects between requests, *e.g.*, if one out of 100 requests arrives early, another one will likely arrive late and the relative disturbance will be minor. Moreover, the relative gap of both dynamic policies as a function of  $n$  reaches a maximum and then decreases as  $n$  grows. This confirms their effectiveness for large  $n$ . Also, all four heuristics’ gaps increase as a function of  $r$ ; the level of flexibility translates into solution complexity for our heuristics. Additionally, the gap tends to increase with  $\ell$ ; this is likely related to an increase in the problem’s complexity. Finally, the ALP-based policy has an average gap smaller than DAP. For less flexible instances ( $r = 1$ ) both approaches average a relative gap of 3.4%, but when the variability and recourse flexibility increases to  $r = 2$  and  $r = 3$  it improves over DAP, from 7.3% to 7.1% for  $r = 2$  and from 9.0% to 8.8% for  $r = 3$ . Although small, this improvement was consistently observed across all instances.

### 6.3 Design of Instance Set 2: Uniform Arrivals

The previous arrival distributions defined by a single parameter could be hiding interesting interdependencies between mean, variance, probability of arrival, and degree of dynamism. We defined a second set of experiments with a fixed number of requests ( $n = 20$ ), waves ( $T = 30$ ) and maximum location ( $\ell = 10$ ). The distance vector  $\mathbf{d}$  and penalty vector  $\mathbf{p}$  are set as in the previous experiments, but arrivals have distributions with a probability  $w$  of arrival at the beginning of the horizon (*i.e.*, the degree of dynamism), a probability  $q$  of not showing up, and a discrete uniform probability  $\frac{1-w-q}{2v+1}$  of arriving during the operation at wave  $t = \max\{1, \mu_i - v\}, \dots, \min\{T - 1, \mu_i + v\}$ , where  $\mu_i$  is a request-dependent parameter drawn i.i.d. from a discrete uniform distribution  $U(0, T - 1)$  for each  $i \in N$ . The parameter  $v$  represents the arrival variability (half of the arrival range). We created 20 instances for each set of parameters  $(v, q, w)$  in the set  $\{(v, q, w) : v = \{0, 1, 2, 4, 8, 30\}, q = \{0, 0.2, 0.4\}, w = \{0, 0.2, 0.4, 0.6, 0.8, 1\} : r + q \leq 1\}$ .

### 6.4 Results for Instance Set 2

Table 4: Overall performance of heuristics in Instance Set 2.

Upper Bound	%GAP	Time per sample-instance (secs)
AP	7.54%	0.0006
APR	5.62%	0.0006
DAP	4.46%	0.0066
DALP	4.24%	0.442

Table 4 presents overall results for each heuristic over the second set of experiments. We notice that our

simple recourse rules in APR capture  $58\% = \frac{7.54-5.62}{7.54-4.24}$  of the total gap improvement that the best dynamic heuristic captures over the static solution AP. Figure 7 presents average relative gaps over instances with different settings of parameters  $q - w$  or  $v$ .

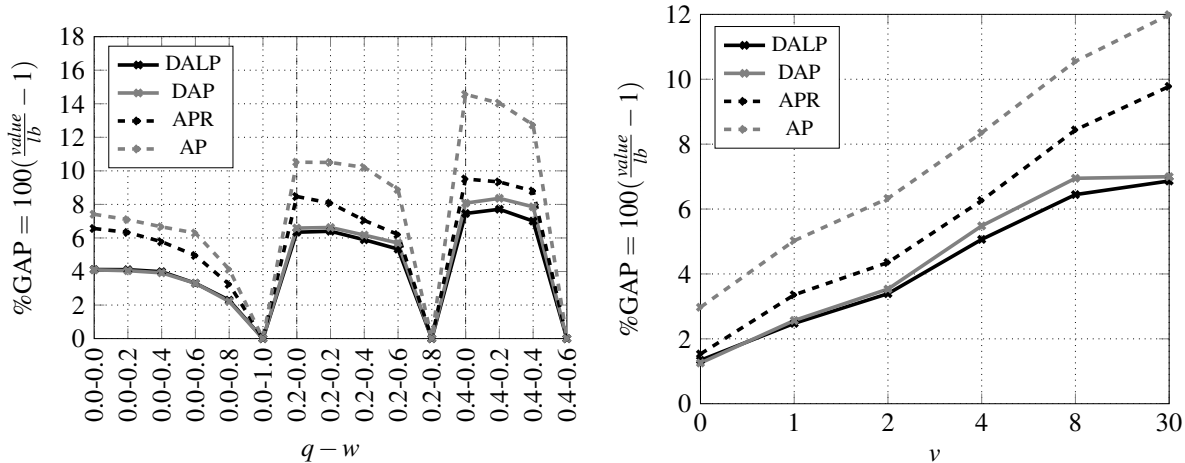


Figure 7: Average percentage gap between heuristics cost and lower bound in Instance Set 2.

From these graphs we conclude that the relative gaps of all four policies decrease as  $w$  increases; the more information available at the initial wave, the closer we can get to a deterministic problem. There is zero gap in the extreme deterministic cases ( $w + q = 1$ ). The value of dynamic solutions also decreases when  $w$  increases, which is expected, since a smaller  $w$  implies a larger degree of dynamism and more importance is placed on recourse actions. Regarding the request arrival probability, the gap increases with  $q$  (unless  $w + q = 1$ ). This means that it is harder to optimize an instance for which there is a bigger probability of no arrival. The value of dynamic solutions also grows with  $q$ . With respect to the variability of the instance, the gap increases as  $v$  increases. This may be due both to a decrease in the lower bound's quality and to an increase in the optimal expected cost. Finally, the dynamic heuristics yield larger marginal costs savings when  $v$  increases. This means that the more variability the system has, the more important it is to implement a dynamic solution. There is also a range of intermediate variability for which DALP clearly dominates DAP. In this range, the additional complexity of ALP yields the most benefit. Table 5 provides four examples of instance families within this range; their average percent reduction in relative gap of DALP over DAP is 15.0%.

Table 5: Average gap percent reduction of DALP for cases with intermediate variability in Instance Set 2.

Family $(q, w, v)$	DALP %GAP	DAP %GAP	% reduction over DAP
(0.4, 0.2, 4)	8.23%	9.67%	14.9%
(0.4, 0.2, 8)	10.79%	12.41%	13.1%
(0.4, 0.4, 4)	7.46%	8.95%	16.6%
(0.4, 0.4, 8)	8.68%	10.31%	15.8%
<b>Aggregate</b>	<b>8.79%</b>	<b>10.34%</b>	<b>15.0%</b>

## 7 Conclusions

We have formulated the dynamic dispatch waves problem (DDWP) to capture the basic aspects of dispatch and routing decisions for same-day delivery. This paper initiates work on the DDWP by studying the single-vehicle stochastic case where customer destinations are placed over the line.

We develop a set of tractable solution policies that differ in their solution dynamism, from an *a priori* solution to fully dynamic policies. Our computational experiments indicate that the performance of an *a priori* policy is good, especially when we include heuristic improvements. In computational tests over two instance sets this policy yields an expected cost within 9.24% and 5.62% of the best lower bound. Nevertheless, we prove that the benefit of a fully dynamic policy can be unbounded in the worst-case scenario. Accordingly, we proposed and experimentally tested two dynamic policies that differ by the nature of the approximate cost-to-go function: the rollout of the *a priori* solution and an ALP-based dynamic policy. The rollout of the *a priori* policy computes this policy at the start of the horizon, but only implements the first action, then updates all known information and re-computes a new *a priori* solution. In both sets of instances it cuts the *a priori* policy’s gap by 28.7% and 20.6%, respectively. We have also found that a dynamic policy that incorporates the ALP approach yields the best possible results. Its marginal improvement as gap reduction for both sets of experiments is 2.6% and 4.9%, respectively. In instance families with intermediate variability, this gap reduction grows to 15.0%.

A final conclusion of our study concerns the relative value of dynamic policies. With all other things being equal, the benefit of a dynamic policy over the optimal *a priori* solution eventually decreases as  $n$  grows, *i.e.*, as the number of potential orders increases. This is unsurprising, since for larger numbers of potential orders one would expect an averaging effect. We found the maximum benefit in dynamic policies for order sets of around 20 to 50; for smaller numbers, the exact optimal solution is still tractable, whereas for larger numbers the *a priori* policy is close to optimality. Many same-day delivery applications, such as grocery home delivery, might expect maximum daily order volume around these numbers. Similarly,

dynamic policies' benefits decrease as orders become more likely to appear at the start of the horizon. In other words, if many of the orders are not placed in the same day at all, but rather are carried over from the previous day, an *a priori* policy performs quite well. It is precisely in the most uncertain environments, where orders can appear at any moment, that new models such as ours offer the most benefit.

Future work on the DDWP needs to consider the solution on a general network topology, and thus become more applicable for SDD operations in urban networks. This problem is quite challenging; in addition to dispatch decisions, it needs to deal with difficult vehicle routing problems. Given this additional difficulty, one could deal with this problem by designing heuristics based on insights from the one-dimensional case. It would also be interesting to extend this model to multiple vehicles that could pool the risk associated with leaving orders unattended and therefore reduce costs. Other extensions could be incorporating vehicle service times at each location or including customer service time windows instead of a deadline at the end of the day. In general, same-day delivery offers many new challenges to the logistics research community.

## References

- [1] D. Adelman, *A Price-Directed Approach to Stochastic Inventory/Routing*, *Operations Research* **52** (2004), 499–514.
- [2] 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.
- [3] 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.
- [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*, working paper, University of Brescia (2015).
- [8] 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.
- [9] D. Bertsekas, *Dynamic programming and optimal control*, vol. 1, Athena Scientific Belmont, MA, 1995.

- [10] D. Bertsekas and D. Castañón, *Rollout algorithms for stochastic scheduling problems*, Journal of Heuristics **5** (1999), no. 1, 89–108.
- [11] D. Bertsimas, *A vehicle routing problem with stochastic demand*, Operations Research **40** (1992), no. 3, 574–585.
- [12] D. Brown, J. Smith, and P. Sun, *Information relaxations and duality in stochastic dynamic programs*, Operations research **58** (2010), 785–801.
- [13] Y. Bukchin, E. Khmel'nitsky, and P. Yakuel, *Optimizing a dynamic order-picking process*, European Journal of Operational Research **219** (2012), no. 2, 335–346.
- [14] 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.
- [15] ———, *Probabilistic traveling salesman problem with deadlines*, Transportation Science **42** (2008), no. 1, 1–21.
- [16] D. Cattaruzza, N. Absi, and D Feillet, *The multi trip vehicle routing problem with time windows and release dates*, Transportation Science, forthcoming (2015).
- [17] E. Çeven and K.R. Gue, *Optimal wave release times for order fulfillment systems with deadlines*, working paper, University of Louisville (2015).
- [18] T. Cheong and C. White, *Dynamic traveling salesman problem: Value of real-time traffic information*, IEEE Transactions on Intelligent Transportation Systems **13** (2012), no. 2.
- [19] J.F. Cordeau, G. Laporte, M. Savelsbergh, and D. Vigo, *Vehicle routing*, Transportation, handbooks in operations research and management science **14** (2006), 367–428.
- [20] D.P. de Farias and B. van Roy, *The linear programming approach to approximate dynamic programming*, Operations Research **51** (2003), no. 6, 850–865.
- [21] V. Desai, V. Farias, and C. Moallemi, *Bounds for Markov decision processes*, Reinforcement Learning and Approximate Dynamic Programming for Feedback Control (2011), 452–473.
- [22] S. Emde and N. Boysen, *One-dimensional vehicle scheduling with a front-end depot and non-crossing constraints*, OR Spectrum **36** (2014), no. 2, 381–400.
- [23] G. Erdoğan, M. Battarra, and G. Laporte, *Scheduling twin robots on a line*, Naval Research Logistics (NRL) **61** (2014), no. 2, 119–130.
- [24] 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.
- [25] N. Gademann and S. Velde, *Order batching to minimize total travel time in a parallel-aisle warehouse*, IIE transactions **37** (2005), no. 1, 63–75.
- [26] M. Gendreau, G. Laporte, and R. Séguin, *Stochastic vehicle routing*, European Journal of Operational Research **88** (1996), no. 1, 3–12.



- [27] G. Godfrey and W. Powell, *An adaptive dynamic programming algorithm for dynamic fleet management, i: Single period travel times*, *Transportation Science* **36** (2002), no. 1, 21–39.
- [28] B.L. Golden, S. Raghavan, and E.A. Wasil (eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer, 2008.
- [29] J. Goodson, J. Ohlmann, and B. 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.
- [30] G. Gutin and A.P. Punnen (eds.), *The Traveling Salesman Problem and Its Variations*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.
- [31] Amazon.com Inc., *Annual report form (10-k)*, (2013).
- [32] P. Jaillet, *Probabilistic traveling salesman problems*, Ph.D. thesis, Massachusetts Institute of Technology, 1985.
- [33] ———, *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.
- [34] P. Jaillet and M. Wagner, *Online vehicle routing problems: A survey*, in Golden et al. [28], pp. 221–237.
- [35] A. Kenyon and D. Morton, *Stochastic vehicle routing with random travel times*, *Transportation Science* **37** (2003), no. 1, 69–82.
- [36] G. Laporte, F. Louveaux, and H. Mercure, *The vehicle routing problem with stochastic travel times*, *Transportation science* **26** (1992), no. 3, 161–170.
- [37] ———, *A priori optimization of the probabilistic traveling salesman problem*, *Operations Research* **42** (1994), no. 3, 543–549.
- [38] 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.
- [39] C. Lee, K. Lee, and S. Park, *Robust vehicle routing problem with deadlines and travel time/demand uncertainty*, *Journal of the Operational Research Society* **63** (2011), no. 9, 1294–1306.
- [40] T. Leipälä, *On the solutions of stochastic traveling salesman problems*, *European Journal of Operational Research* **2** (1978), no. 4, 291–297.
- [41] X. Li, P. Tian, and S. Leung, *Vehicle routing problems with time windows and stochastic travel and service times: models and algorithm*, *International Journal of Production Economics* **125** (2010), no. 1, 137–145.
- [42] T. Ma and P. Zhao, *A review of algorithms for order batching problem in distribution center*, *International Conference on Logistics Engineering, Management and Computer Science (LEMCS 2014)*, Atlantis Press, 2014.
- [43] C. Novoa and R. Storer, *An approximate dynamic programming approach for the vehicle routing problem with stochastic demands*, *European Journal of Operational Research* **196** (2009), no. 2, 509–515.

- [44] 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.
- [45] W. Powell, *Approximate dynamic programming: Solving the curses of dimensionality*, John Wiley & Sons, 2007.
- [46] H. Psaraftis, *Dynamic vehicle routing: Status and prospects*, Annals of Operations Research **61** (1995), no. 1, 143–164.
- [47] M. Puterman, *Markov decision processes: discrete stochastic dynamic programming*, John Wiley & Sons, 2009.
- [48] P. Schweitzer and A. Seidmann, *Generalized polynomial approximations in markovian decision processes*, Journal of mathematical analysis and applications **110** (1985), no. 2, 568–582.
- [49] N. Secomandi and F. Margot, *Reoptimization approaches for the vehicle-routing problem with stochastic demands*, Operations Research **57** (2009), no. 1, 214–230.
- [50] V. Sehgal, *Online retail forecast (US), 2013 to 2018*, Technical report, Forrester Research (2014).
- [51] D. Taş, N. Dellaert, T. van Woensel, and T. de Kok, *Vehicle routing problem with stochastic travel times including soft time windows and service costs*, Computers & Operations Research (2012).
- [52] B. Thomas, *Dynamic vehicle routing*, Wiley Encyclopedia of Operations Research and Management Science (2010).
- [53] A. Toriello, W.B. Haskell, and M. Poremba, *A Dynamic Traveling Salesman Problem with Stochastic Arc Costs*, Operations Research **62** (2014), 1107–1125.
- [54] 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.
- [55] 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.
- [56] C. Zhang, Y. Wan, J. Liu, and R.J. Linn, *Dynamic crane deployment in container storage yards*, Transportation Research Part B: Methodological **36** (2002), no. 6, 537–555.

## Appendix

### Proof of Proposition 5.1

*Proof.* Applying restriction (6) to (2) yields the LP

$$C^{ALP} = \max_{\{\mathbf{a}, \mathbf{b}, \mathbf{v}\}} \mathbb{E}_{R_T} \left[ \sum_{i \in \hat{R}} a_i^T + \sum_{i \in N \setminus \hat{R}} b_i^T \right] - \sum_{t=1}^T v_t \quad (10a)$$

$$\text{s.t. } \sum_{i \in R} a_i^0 + \sum_{i \in P} b_i^0 = \sum_{i \in R} p_i, \quad \forall (R, P) \in \Xi \quad (10b)$$

$$\sum_{i \in R} a_i^t + \sum_{i \in P} b_i^t - \mathbb{E}_{F_1^t} \left[ \sum_{i \in R \cup F_1^t} a_i^{t-1} + \sum_{i \in P \setminus F_1^t} b_i^{t-1} \right] \leq v_t, \quad \forall t \in \mathcal{T}, (R, P) \in \Xi \quad (10c)$$

$$\sum_{i \in R} a_i^t + \sum_{i \in P} b_i^t - \mathbb{E}_{F_d^t} \left[ \sum_{i \in R_d \cup F_d^t} a_i^{t-d} + \sum_{i \in P \setminus F_d^t} b_i^{t-d} \right] \leq \alpha d + \sum_{k=t-d+1}^t v_k, \quad \forall t \in \mathcal{T}, (R, P) \in \Xi, d \in \mathcal{A}_R^t. \quad (10d)$$

Model (10) has a polynomial number of variables for a given  $n$  and  $T$ , but it has exponentially many terms within the expectations and constraints. We prove Proposition 5.1 in two steps. First, we compute a closed form for the expectations in model (10). Then we show a one to one equivalence between both domains.

The expectations in (10) are given by

$$\begin{aligned} \mathbb{E}_{R_T} \left[ \sum_{i \in \hat{R}} a_i^T + \sum_{i \in N \setminus \hat{R}} b_i^T \right] &= \sum_{i \in N} (\mathbb{P}(\tau_i = T) a_i^T + \mathbb{P}(\tau_i < T) b_i^T) \\ \mathbb{E}_{F_1^t} \left[ \sum_{i \in R \cup F_1^t} a_i^{t-1} + \sum_{i \in P \setminus F_1^t} b_i^{t-1} \right] &= \sum_{i \in R} a_i^{t-1} + \sum_{i \in P} f_{it} a_i^{t-1} + \bar{f}_{it} b_i^{t-1} \\ \mathbb{E}_{F_d^t} \left[ \sum_{i \in R_d \cup F_d^t} a_i^{t-d} + \sum_{i \in P \setminus F_d^t} b_i^{t-d} \right] &= \sum_{i \in R_d} a_i^{t-d} + \sum_{i \in P} g_{it}^d a_i^{t-d} + \bar{g}_{it}^d b_i^{t-d}. \end{aligned}$$

Replacing them in (10) yields

$$\max_{\{\mathbf{a}, \mathbf{b}, \mathbf{v} \geq 0\}} \sum_{i \in N} (\mathbb{P}(\tau_i = T) a_i^T + \mathbb{P}(\tau_i < T) b_i^T) - \sum_{t=1}^T v_t \quad \text{s.t.} \quad (11a)$$

$$\sum_{i \in R} a_i^0 + \sum_{i \in P} b_i^0 = \sum_{i \in R} p_i \quad \forall (R, P) \in \Xi \quad (11b)$$

$$\sum_{i \in R} (a_i^t - a_i^{t-1}) + \sum_{i \in P} (b_i^t - f_{it} a_i^{t-1} - \bar{f}_{it} b_i^{t-1}) \leq v_t \quad \forall t \in \mathcal{T}, \forall (R, P) \in \Xi \quad (11c)$$

$$\sum_{i \in R_d} a_i^t + \sum_{i \in R_d} (a_i^t - a_i^{t-d}) + \sum_{i \in P} (b_i^t - g_{it}^d a_i^{t-d} - \bar{g}_{it}^d b_i^{t-d}) \leq \sum_{k=t-d+1}^t v_k + \alpha d \quad \forall t \in \mathcal{T}, \forall (R, P) \in \Xi, \forall d \in \mathcal{A}_R^t, \quad (11d)$$

where we still have an exponential number of constraints. We prove that (7) is equivalent to (11) by showing equality between both domains.

1. (7b)  $\iff$  (11b): Suppose that  $(\mathbf{a}, \mathbf{b}, \mathbf{v})$  satisfies (11b). If  $R = \{i\}$  and  $P = \emptyset$  we get  $a_i^t = p_i$ , and if  $R = \emptyset$  and  $P = \{i\}$  we get  $b_i^t = 0$ . Now, suppose that  $(\mathbf{a}, \mathbf{b}, \mathbf{v})$  satisfies (7b) and add  $a_i = p_i$  and  $b_j = 0$  over any feasible pair of sets  $(R, P) \in \Xi$  to get (11b).
2. (7c), (7d), (7e)  $\iff$  (11c): Suppose that  $(\mathbf{a}, \mathbf{b}, \mathbf{v})$  satisfies (11c). For each  $t \in \mathcal{T}$ , choose a particular  $(R, P) \in \Xi$  as follows: put  $i \in R$  if  $a_i^t - a_i^{t-1} > \max\{0, b_i^t - f_{ii}a_i^{t-1} - \bar{f}_{ii}b_i^{t-1}\}$ , and put  $i \in P$  if  $b_i^t - f_{ii}a_i^{t-1} - \bar{f}_{ii}b_i^{t-1} > \max\{0, a_i^t - a_i^{t-1}\}$ .

Then, for  $(i, t)$  set  $s_{it} = \max\{0, a_i^t - a_i^{t-1}, b_i^t - f_{ii}a_i^{t-1} - \bar{f}_{ii}b_i^{t-1}\}$  and we get

$$v_t \geq \sum_{i \in R} a_i^t - a_i^{t-1} + \sum_{i \in P} b_i^t - f_{ii}a_i^{t-1} - \bar{f}_{ii}b_i^{t-1} = \sum_{i \in N} \max\{0, a_i^t - a_i^{t-1}, b_i^t - f_{ii}a_i^{t-1} - \bar{f}_{ii}b_i^{t-1}\} = \sum_{i \in N} s_{it}.$$

Now suppose that  $(\mathbf{a}, \mathbf{b}, \mathbf{v}, \mathbf{s})$  satisfies (7c), (7d), (7e), select any pair  $(R, P) \in \Xi$  and we have

$$v_t \geq \sum_{i \in N} s_{it} \geq \sum_{i \in R} s_{it} + \sum_{i \in P} s_{it} \geq \sum_{i \in R} (a_i^t - a_i^{t-1}) + \sum_{i \in P} (b_i^t - f_{ii}a_i^{t-1} - \bar{f}_{ii}b_i^{t-1}).$$

3. (11d)  $\iff$  (7f), (7g), (7h). Consider that  $(\mathbf{a}, \mathbf{b}, \mathbf{v})$  satisfies (11d). For each  $t \in \mathcal{T}$  and  $d \in \mathcal{A}_N^t$ , choose  $(R, P) \in \Xi$  as follows: put  $i \in R$  if  $a_i^t - \mathbb{I}_{d_i > d} a_i^{t-d} > \max\{0, b_i^t - g_{ii}^d a_i^{t-d} - \bar{g}_{ii}^d b_i^{t-d}\}$ , and put  $i \in P$  if  $b_i^t - g_{ii}^d a_i^{t-d} - \bar{g}_{ii}^d b_i^{t-d} > \max\{0, a_i^t - \mathbb{I}_{d_i > d} a_i^{t-d}\}$ .

Then, for each  $(i, t, d)$  define  $u_{it}^d = \max\{0, a_i^t - \mathbb{I}_{d_i > d} a_i^{t-d}, b_i^t - g_{ii}^d a_i^{t-d} - \bar{g}_{ii}^d b_i^{t-d}\}$ . By (11d), we get

$$\begin{aligned} \alpha d + \sum_{k=t-d+1}^t v_k &\geq \sum_{i \in R} (a_i^t - \mathbb{I}_{d_i > d} a_i^{t-d}) + \sum_{i \in P} (b_i^t - g_{ii}^d a_i^{t-d} - \bar{g}_{ii}^d b_i^{t-d}) \\ &= \sum_{i \in N} \max\{0, a_i^t - \mathbb{I}_{d_i > d} a_i^{t-d}, b_i^t - g_{ii}^d a_i^{t-d} - \bar{g}_{ii}^d b_i^{t-d}\} = \sum_{i \in N} u_{it}^d. \end{aligned}$$

Now, suppose that  $(\mathbf{a}, \mathbf{b}, \mathbf{v}, \mathbf{u})$  satisfies (7f), (7g), (7h), select any pair  $(R, P) \in \Xi$  and get

$$\begin{aligned} \alpha d + \sum_{k=t-d+1}^t v_k &\geq \sum_{i \in N} u_{it}^d \\ &\geq \sum_{i \in R} u_{it}^d + \sum_{i \in P} u_{it}^d \\ &\geq \sum_{i \in R} (a_i^t - \mathbb{I}_{d_i > d} a_i^{t-d}) + \sum_{i \in P} (b_i^t - g_{ii}^d a_i^{t-d} - \bar{g}_{ii}^d b_i^{t-d}) \\ &= \sum_{i \in R_d} a_i^t + \sum_{i \in \bar{R}_d} (a_i^t - a_i^{t-d}) + \sum_{i \in P} (b_i^t - g_{ii}^d a_i^{t-d} - \bar{g}_{ii}^d b_i^{t-d}). \quad \square \end{aligned}$$

## 7.1 Proof of Property 5.2

*Proof.* We start proving that there exists at least one optimal solution for (7) satisfying  $a_i^t \leq p_i$  and  $b_i^t \leq g_{ii}^t p_i$  for all  $i \in N$  and  $t \in \mathcal{T}_0$ . Choose any  $i \in N$  and do forward induction on  $t$ .

- $t = 0$  is given by constraints (7b).

- Inductive step:

Assume that  $a_i^k \leq p_i$  and that  $b_i^k \leq g_{ik}^k p_i$  for all  $k < t$ . We prove the statement for step  $t$ .

Suppose that  $a_i^t = p_i + \delta_a$  and  $b_i^t = g_{it}^t p_i + \delta_b$ , with  $\varepsilon = \max\{\delta_a, \delta_b\} > 0$ . By the inductive hypothesis, (7c),(7d), (7f) and (7g) it implies that  $s_{it} \geq \varepsilon$ ,  $u_{it}^d \geq \varepsilon$ , for all  $d \in \mathcal{A}_N^t$  and by (7e) we have  $v_t \geq \varepsilon$ .

So, update the variables for time  $t$  as follows:  $a_i^t \leftarrow p_i$ ;  $b_i^t \leftarrow b_i^t - \varepsilon$ ;  $s_{it} \leftarrow s_{it} - \varepsilon$ ;  $u_{it}^d \leftarrow u_{it}^d - \varepsilon$ ,  $\forall d \in \mathcal{A}_N^t$  and  $v_t \leftarrow v_t - \varepsilon$ . Also, update the variables for time  $v > t$ :  $a_i^v \leftarrow a_i^v - \varepsilon$  and  $b_i^v \leftarrow b_i^v - \varepsilon$ . These changes keep (7) feasible and the objective value does not changes (the reduction in  $v_t$  increases the objective by  $\varepsilon$ , but the change in  $\mathbb{P}(\tau_i = T)a_i^T + \mathbb{P}(\tau_i < T)b_i^T$  reduces it by  $\varepsilon$ ).

Now, let us show that there exists at least one optimal solution for (7) satisfying  $a_i^t \geq 0$ ,  $b_i^t \geq 0$  for all  $i \in N$  and  $t \in \mathcal{T}_0$ . Choose any  $i \in N$  and do forward induction on  $t$ .

- $t = 0$  is given by constraints (7b).

- Inductive step:

Assume that  $a_i^k \geq 0$ ,  $b_i^k \geq 0$  for all  $k < t$ . We prove the statement for step  $t$ .

Suppose that:  $a_i^t < 0$  and/or  $b_i^t < 0$ . We can set these variables equal to 0 without losing feasibility. The objective does not changes unless  $t = T$ , and for this last case it improves. □

## 7.2 Proof of Property 5.3

*Proof.* Choose any  $i \in N$ . We prove by induction on  $t$  that there exists an optimal solution satisfying  $a_i^t = p_i$  and  $b_i^t = g_{it}^t p_i$  for all  $i \in N, t \in \mathcal{T}_0 : d_i > t$ .

- $t = 0$  is given by (7b).

- Inductive step:

Assume that  $a_i^k = p_i$ ,  $b_i^k = g_{ik}^k p_i, \forall k \in \mathcal{T}_0 : d_i > k$  with  $k < t$  and suppose that the optimal solution is such that  $a_i^t = p_i - \delta_a, b_i^t = g_{it}^t p_i - \delta_b$ , where  $\max\{\delta_a, \delta_b\} > 0$ . We can reassign these two variables, i.e.,  $a_i^t \leftarrow p_i$  and  $b_i^t \leftarrow g_{it}^t p_i$ , keeping feasibility and without reducing the objective value. Just note that for constraints (7f) we have  $d_i > d$  (given by  $d_i > t$  and  $d \in \mathcal{A}_N^t$ ). Thus, all constraints involving the reassigned variables are

$$\begin{aligned}
s_{it} &\geq a_i^t - a_i^{t-1} = -\delta_a \\
s_{it} &\geq b_i^t - f_{it} a_i^{t-1} - \bar{f}_{it} b_i^{t-1} = -\delta_b \\
s_{it+1} &\geq a_i^{t+1} - a_i^t = a_i^{t+1} - p_i + \delta_a \\
s_{it+1} &\geq b_i^{t+1} - f_{it+1} a_i^t - \bar{f}_{it+1} b_i^t = b_i^{t+1} - g_{it+1}^{t+1} p_i + f_{it+1} \delta_a + \bar{f}_{it+1} \delta_b \\
u_{it}^d &\geq a_i^t - a_i^{t-d} = -\delta_a \\
u_{it}^d &\geq b_i^t - g_{it}^d a_i^{t-d} - \bar{g}_{it}^d b_i^{t-d} = -\delta_b \\
u_{it+d}^d &\geq a_i^{t+d} - a_i^t = a_i^{t+d} - p_i + \delta_a \\
u_{it+d}^d &\geq b_i^{t+d} - g_{it+d}^d a_i^t - \bar{g}_{it+d}^d b_i^t = b_i^{t+d} - g_{it+d}^{t+d} p_i + g_{it+d}^d \delta_a + \bar{g}_{it+d}^d \delta_b,
\end{aligned}$$

and when  $\delta_a, \delta_b \rightarrow 0$  the lower bounds for  $\mathbf{u}$  and  $\mathbf{s}$  do not increase, since  $\mathbf{u}$  and  $\mathbf{s}$  are nonnegative. The missing case, i.e.  $b_i^t = g_{it}^t p_i$  when  $d_i = t$  follows a similar proof.

□

### 7.3 Proof of Theorem 5.4

For this proof we simplify our formulation to keep the intuition as simple as possible. The action set  $\mathcal{A}_R^t$  in state  $(t, R, P)$  will be  $\{d \in \mathbb{Z}_+ : d \leq t\}$ , and so, will include possibly suboptimal actions. So, consider the stochastic DDWP

$$\begin{aligned}
C^* &= \max_C \mathbb{E}_{\hat{R}} [C_T(\hat{R}, N \setminus \hat{R})] \\
\text{s.t. } C_0(R, P) &\leq \sum_{i \in R} P_i, & (R, P) \in \Xi \\
C_t(R, P) &\leq \mathbb{E}_{F_t^i} [C_{t-1}(R \cup F_t^i, P \setminus F_t^i)], & t \in \mathcal{T}, (R, P) \in \Xi \\
C_t(R, P) &\leq \alpha d + \mathbb{E}_{F_d^i} [C_{t-d}(R_d \cup F_d^i, P \setminus F_d^i)], & t \in \mathcal{T}, d \in \mathbb{Z}_+ : d \leq t, (R, P) \in \Xi,
\end{aligned}$$

and its ALP bound

$$\begin{aligned}
C' &= \max_{a, v, s, u} \sum_{i \in N} (\mathbb{P}(\tau_i = T) a_i^T + \mathbb{P}(\tau_i < T) b_i^T) - \sum_{k=1}^T v_k \\
\text{s.t. } a_i^0 &\leq p_i, b_i^0 \leq 0, & i \in N \\
a_i^t - a_i^{t-1} - s_{it} &\leq 0, & i \in N, t \in \mathcal{T} \\
b_i^t - f_{it} a_i^{t-1} - \bar{f}_{it} b_i^{t-1} - s_{it} &\leq 0, & i \in N, t \in \mathcal{T} \\
\sum_{i \in N} s_{it} - v_t &\leq 0, & t \in \mathcal{T} \\
a_i^t - u_{it}^d &\leq 0, & i \in N, t \in \mathcal{T}, d \in \{d_i, \dots, t\} \\
a_i^t - a_i^{t-d} - u_{it}^d &\leq 0, & i \in N, t \in \mathcal{T}, d \in \{1, \dots, \min(d_i - 1, t)\} \\
b_i^t - g_{it}^d a_i^{t-d} - \bar{g}_{it}^d b_i^{t-d} - u_{it}^d &\leq 0, & i \in N, t \in \mathcal{T}, d \in \{1, \dots, t\} \\
\sum_{i \in N} u_{it}^d - \sum_{k=t-d+1}^t v_k &\leq \alpha d, & t \in \mathcal{T}, d \in \{1, \dots, t\} \\
\mathbf{s}, \mathbf{u} &\geq 0.
\end{aligned}$$

For the deterministic case we get  $\mathbb{P}(\tau_i = T) = \mathbb{I}_{(\tau_i=T)}$ ,  $f_{it} = \mathbb{I}_{(\tau_i=t-1)}$  and  $g_{it}^d = \mathbb{I}_{(t-d \leq \tau_i < t)}$ . The ALP collapses to

$$\begin{aligned}
C' &= \max_{a, v, s, u} \sum_{i \in N} (a_i^T \mathbb{I}_{(\tau_i=T)} + b_i^T \mathbb{I}_{(\tau_i < T)}) - \sum_{k=1}^T v_k \\
\text{s.t. } a_i^0 &\leq 0, b_i^0 \leq 0, & i \in N \\
a_i^t - a_i^{t-1} - s_{it} &\leq 0, & i \in N, t \in \mathcal{T} \\
b_i^t - b_i^{t-1} - \mathbb{I}_{(t-1 \leq \tau_i < t)} (a_i^{t-1} - b_i^{t-1}) - s_{it} &\leq 0, & i \in N, t \in \mathcal{T} \\
\sum_{i \in N} s_{it} - v_t &\leq 0, & t \in \mathcal{T} \\
a_i^t - u_{it}^d &\leq 0, & i \in N, t \in \mathcal{T}, d \in \{d_i, \dots, t\} \\
a_i^t - a_i^{t-d} - u_{it}^d &\leq 0, & i \in N, t \in \mathcal{T}, d \in \{1, \dots, \min(d_i - 1, t)\} \\
b_i^t - b_i^{t-d} - \mathbb{I}_{(t-d \leq \tau_i < t)} (a_i^{t-d} - b_i^{t-d}) - u_{it}^d &\leq 0, & i \in N, t \in \mathcal{T}, d \in \{1, \dots, t\} \\
\sum_{i \in N} u_{it}^d - \sum_{k=t-d+1}^t v_k &\leq \alpha d, & t \in \mathcal{T}, d \in \{1, \dots, t\}
\end{aligned}$$

$$\mathbf{s}, \mathbf{u} \geq 0.$$

From this point we assume without loss of generality that  $d_i \leq \tau_i$ . Otherwise, we can transform the model to an equivalent one satisfying this requirement. If request  $i$  does not arrive ( $\tau_i < 0$ ), the optimal ALP value does not get altered by removing it, since at optimality  $a_i^t = b_i^t = 0, \forall t \in \mathcal{T}_0$ . In case that  $0 < \tau_i < d_i$ , i.e. the order arrives but cannot be served, one optimal solution is  $a_i^t = p_i, \forall t \in \mathcal{T}_0$  and may be removed from the analysis by adding a constant  $p_i$  to the objective.

Now, let us preset some variables in the ALP:

- $a_i^t = 0$ , for all  $i \in N, t > \tau_i$ ; the open order cost before arrival is zero.
- $b_i^t = a_i^{\tau_i}$ , for all  $i \in N, t > \tau_i$ ; the potential order cost before arrival is equal to the open order cost upon arrival.
- $b_i^t = 0$ , for all  $i \in N, t \leq \tau_i$ , i.e., the potential order cost after arrival is zero.

The remaining model is still an underestimate of  $C^*$  given by

$$\begin{aligned}
C'' &= \max_{a,v,s,u} \sum_{i \in N} a_i^{\tau_i} - \sum_{k=1}^T v_k \\
\text{s.t. } (w) \quad &a_i^t - a_i^{t-1} - s_{it} \leq 0, && i \in N, t \in \{1, \dots, \tau_i\} \\
(m) \quad &a_i^t - u_{it}^d \leq 0, && i \in N, t \in \{d_i, \dots, \tau_i\}, d \in \{d_i, \dots, t\} \\
(\alpha) \quad &a_i^t - a_i^{t-d} - u_{it}^d \leq 0, && i \in N, d \in \{1, \dots, d_i - 1\}, t \in \{d, \dots, \tau_i\}, \\
(\beta) \quad &a_i^{\tau_i} - a_i^{t-d} - u_{it}^d \leq 0, && i \in N, t \in \{\tau_i + 1, \dots, T\}, d \in \{t - \tau_i, \dots, t\} \\
(\gamma) \quad &a_i^0 \leq p_i, && i \in N \\
(Z) \quad &\sum_{i \in N} s_{it} - v_t \leq 0, && t \in \mathcal{T} \\
(Y) \quad &\sum_{i \in N} u_{it}^d - \sum_{k=t-d+1}^t v_k \leq d, && t \in \mathcal{T}, d \in \{1, \dots, t\} \\
&\mathbf{s}, \mathbf{u} \geq 0,
\end{aligned}$$

and its dual problem is

$$C'' = \min_{Z,Y,\alpha,\beta,\gamma,m,w \geq 0} \sum_{i \in N} p_i \gamma_i + \sum_{t=1}^T \sum_{d=1}^t d Y_{t,d} \quad (18a)$$

$$\text{s.t. } (v) \quad Z_t + \sum_{t'=t}^T \sum_{d=t'-t+1}^{t'} Y_{t',d} = 1, \quad t \in \mathcal{T} \quad (18b)$$

$$(s) \quad w_i^t \leq Z_t, \quad i \in N, t \in \{1, \dots, \tau_i\} \quad (18c)$$

$$(u) \quad m_{i,t}^d \leq Y_{t,d}, \quad i \in N, t \in \{d_i, \dots, \tau_i\}, \\ d \in \{d_i, \dots, t\} \quad (18d)$$

$$\alpha_{i,t}^d \leq Y_{t,d}, \quad i \in N, d \in \{1, \dots, d_i - 1\}, \\ t \in \{d, \dots, \tau_i\} \quad (18e)$$

$$\beta_{i,t}^d \leq Y_{t,d}, \quad i \in N, t \in \{\tau_i + 1, \dots, T\},$$

$$d \in \{t - \tau_i, \dots, t\} \quad (18f)$$

$$(a) \gamma_i = \left( w_i^1 + \sum_{k=1+\tau_i}^T \beta_{i,k}^k + \sum_{k=1}^{d_i-1} \alpha_{i,k}^k \right), \quad \begin{array}{l} i \in N, \\ t = 0 \end{array} \quad (18g)$$

$$\begin{aligned} & \left( w_i^t + \sum_{d=1}^t \alpha_{i,t}^d \right) \\ & = \left( w_i^{t+1} + \sum_{k=1+\tau_i}^T \beta_{i,k}^{k-t} + \sum_{k=1}^{d_i-1} \mathbb{I}_{(k+t \leq \tau_i)} \alpha_{i,k+t}^k \right), \quad \begin{array}{l} i \in N, \\ t \in \{1, \dots, d_i - 1\} \end{array} \end{aligned} \quad (18h)$$

$$\begin{aligned} & \left( w_i^t + \sum_{d=d_i}^t m_{i,t}^d + \sum_{d=1}^{d_i-1} \alpha_{i,t}^d \right) \\ & = \left( w_i^{t+1} + \sum_{k=1+\tau_i}^T \beta_{i,k}^{k-t} + \sum_{k=1}^{d_i-1} \mathbb{I}_{(k+t \leq \tau_i)} \alpha_{i,k+t}^k \right), \quad \begin{array}{l} i \in N, \\ t \in \{d_i, \dots, \tau_i - 1\} \end{array} \end{aligned} \quad (18i)$$

$$\begin{aligned} & \left( w_i^{\tau_i} + \sum_{d=d_i}^{\tau_i} m_{i,\tau_i}^d + \sum_{d=1}^{d_i-1} \alpha_{i,t}^d \right) \\ & + \left( \sum_{k=\tau_i+1}^T \sum_{d=k-\tau_i+1}^k \beta_{i,k}^d \right) = 1, \quad \begin{array}{l} i \in N, \\ t = \tau_i. \end{array} \end{aligned} \quad (18j)$$

Consider (18):

1. First note that (18b) are equivalent to the following network flow balance constraints

$$1 = Z_T + \sum_{d=1}^T Y_{T,d} \quad (19a)$$

$$Z_{t+1} + \sum_{t'=t+1}^T Y_{t',t'-t} = Z_t + \sum_{d=1}^t Y_{t,d}, \quad t \in \{1, \dots, T-1\} \quad (19b)$$

$$Z_1 + \sum_{t'=2}^T Y_{t',t'-1} = 1 \quad (19c)$$

$$Z, Y \geq 0, \quad (19d)$$

represented in Figure 8.

*Proof.* Equivalence is obtained by subtracting constraint  $t$  from constraint  $t+1$  in (18b) for all  $t \in \{T-1, \dots, 2\}$ . The flow balance constraint at node  $t = T$  comes explicitly, and the flow balance constraint at node  $t = 1$  is obtained by adding the previously derived equations.  $\square$

Therefore, substructure (19) has integral extreme points.

2. Now, let us study the remaining constraints. Note that for a given  $(Z, Y)$  the resulting problem in variables  $(\alpha, \beta, \gamma, w, m)$  collapses to  $n$  independent capacitated minimum cost network flow problems (CMCNF) for each order  $i \in N$  defined in (20)

$$\gamma_i(Z, Y) = \min_{\alpha_i, \beta_i, \gamma_i, m_i, w_i \geq 0} \gamma_i \quad (20a)$$

$$\text{s.t. (18c), (18d), (18e), (18f), (18g), (18h), (18i), (18j).} \quad (20b)$$

In this network there is a set of nodes given by  $\{0, \dots, \tau_i\}$  and a sink node  $S^i$  defined by the (redundant) flow balance constraint  $\gamma_i + \sum_{t=d_i}^{\tau_i} \sum_{x=d_i}^t m_{i,t}^d = 1$  obtained when adding (18g), (18h), (18i) and (18j). We



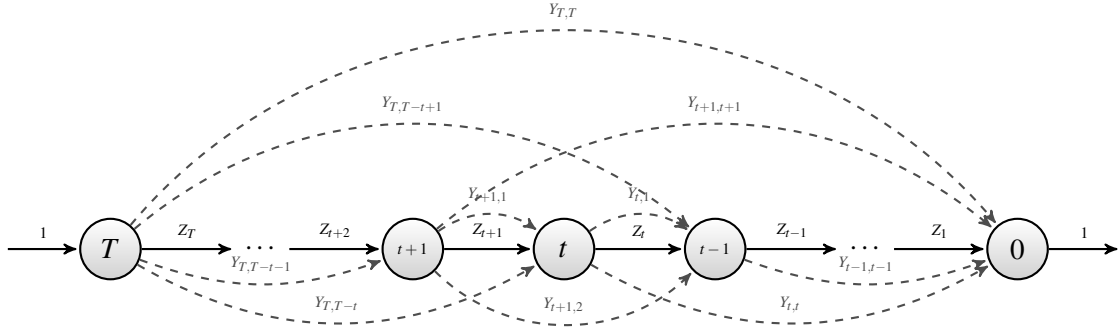


Figure 8: Network Structure in  $(Z, Y)$ -domain

would like to minimize the cost of moving one unit of flow from node  $\tau_i$  to the sink node. There are five types of arcs available in (20) given by

- Type 1 arc ( $\gamma_i$ ) going from node 0 to  $S^i$ . Our objective is to minimize the value of this flow, since it is the only one with non-zero cost.
- Type 2 arcs ( $m_{i,t}^d$ ) going from node  $t \in \{d_i, \dots, \tau_i\}$  to  $S^i$ . We want to maximize these flows, but these arc flows are bounded by  $Y_{t,d}$ .
- Type 3 arcs ( $w_t^t$ ) going from  $t$  to  $t-1$  for each  $t \in \{1, \dots, \tau_i\}$ . These flows are bounded by  $Z_t$ .
- Type 4 arcs ( $\alpha_{i,t}^d$ ) going from a node  $t \in \{1, \dots, \tau_i\}$  to any node  $t-d$  for each  $d < d_i$  and  $d \leq t$ ; also bounded by  $Y_{t,d}$ .
- Type 5 arcs ( $\beta_{i,k}^d$ ) going from node  $\tau_i$  to any node  $t \in \{0, \dots, \tau_i-1\}$  for each  $k \in \mathbb{Z}_+$  and  $d \in \mathbb{Z}_+$  satisfying  $\tau_i < k \leq T$  and  $k-d=t$ ; also bounded by  $Y_{k,d}$ .

Note that problem (20) is feasible for any value  $(Z, Y) \in (19)$ . Its network is graphically represented in Figure 9.

If we put these two comments together, the dual ALP in (18) is equal to

$$\min_{(Z,Y) \in (19)} C(Z, Y) := \underbrace{\sum_{i \in N} p_i \gamma_i(Z, Y)}_{P(Z, Y)} + \underbrace{\sum_{t=1}^T \sum_{d=1}^t d Y_{t,d}}_{C_{OP}(Y)}. \quad (21)$$

We show in two parts that (21) has an optimal value equal to the optimal cost of the deterministic DDWP in (4). First, we prove that any feasible dispatch for the deterministic DDWP has a one-to-one mapping with integer feasible solutions  $(Z, Y)$  to (21). Then, we show that without loss of optimality a solution of (21) can be assumed integral.

**Part 1:** Consider any feasible dispatch with lengths  $\{d^1, \dots, d^K\}$  and dispatch times  $\{t^1, \dots, t^K\}$ . Then, there is a unique integer solution of  $(Z, Y)$  representing this operation. Just set to zero all components of  $Y$  except for  $Y_{t^k, d^k} = 1, \forall k \in \{1, \dots, K\}$  and set  $Z$  to satisfy (18b). Thus,  $Y_{t,d}$  represents a dispatch at  $t$  with distance

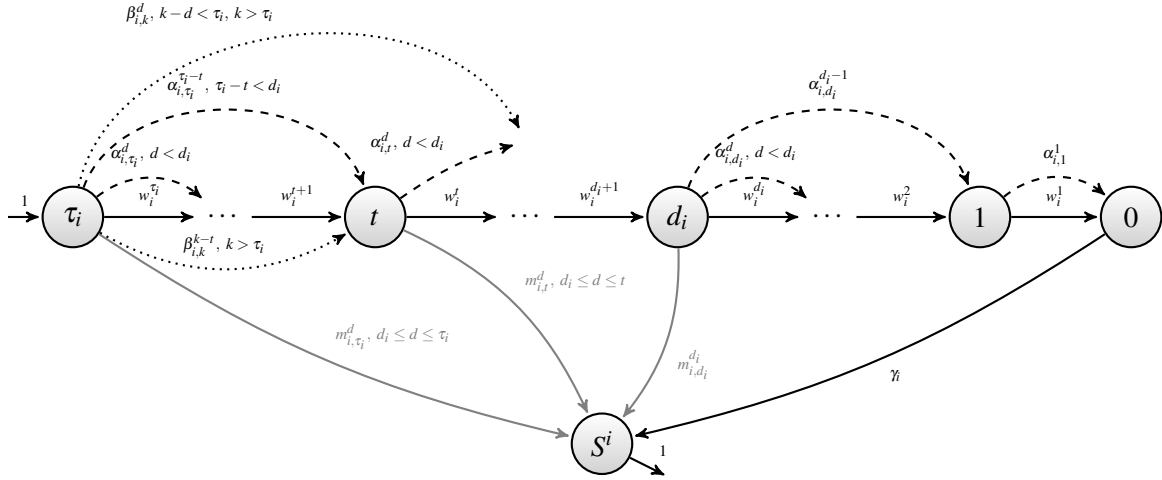


Figure 9: Network for  $i^{\text{th}}$  order subproblem.

length  $d$  and  $Z_t$  represents waiting at the depot between  $t$  and  $t - 1$ . Its corresponding operational dispatch cost matches the second term in the objective of (21), i.e.  $C_{OP}(Y) := \sum_{t=1}^T \sum_{d=1}^t dY_{t,d} = \sum_{k=1}^K d^k$ . Also, let

$$\eta_i = \begin{cases} 1 & \text{if } d^k \geq d_i \text{ and } t^k \leq \tau_i, \text{ for some } k \in \{1, \dots, K\} \\ 0 & \text{otherwise} \end{cases}$$

indicate whether order  $i$  is covered by any dispatch or not. If  $\eta_i = 0$ , then all type 2 arcs for subproblem (20) cannot be used, i.e.  $m_{i,t}^d \leq 0$  for  $d_i \leq t \leq \tau_i$  and  $d_i \leq d \leq t$ , so there is a unique path from  $\tau_i$  to  $S^i$  with  $\gamma_i = 1$ . If  $\eta_i = 1$ , then a new  $(\tau_i - S^i)$ -path arises with capacity one. The idea is to move the unit flow horizontally using type 3 arcs ( $w_t^t = 1$ ) at each node  $t : 1 \leq t \leq \tau_i$  when  $Z_t = 1$ . Otherwise, if  $Z_t = 0$  there are three potential scenarios:

- A dispatch at  $t$  covers  $i$ , i.e.,  $d \geq d_i$ . Then we can use the corresponding type 2 arc  $m_{i,t}^d = Y_{t,d} = 1$  and reach the sink node  $S^i$  at zero cost ( $\gamma_i = 0$ ).
- A dispatch at  $t$  does not cover  $i$ , i.e.  $d < d_i$ . Then we can use the corresponding type 4 arc  $\alpha_{i,t}^d = Y_{t,d} = 1$  and reach node  $t - d$  at zero cost. Since  $\eta_i = 1$ , we proceed until we find the type 2 arc associated with the earliest dispatch that covers  $i$ .
- We have  $t = \tau_i$  and there is a dispatch at time  $k > \tau_i$  with distance  $d$  such that  $k - d < \tau_i$ . Then we can send one unit of flow in a type 5 arc to node  $k - d$ , i.e.,  $b_{i,k}^d = Y_{k,d} = 1$ . Again, we proceed until we find the earliest type 2 arc.

The first cost term in (21) will be exactly equal to the penalties paid for orders left unattended:  $P(Z, Y) := \sum_{i \in N} P_i \gamma_i = \sum_{i \in N} P_i \cdot \sum_{n_i=0}^1$

**Part 2:** Now we prove that without loss of optimality  $Z, Y$  is binary, and hence an optimal solution is an optimal dispatch for the deterministic DDWP. Assume by contradiction that  $Y$  has fractional components

and that  $C(Z, Y) < C(\bar{Z}, \bar{Y})$  for any integral solution  $(\bar{Z}, \bar{Y}) \in (19)$ . We can express  $(Z, Y)$  as a convex combination of the extreme points  $(Z^1, Y^1), \dots, (Z^p, Y^p)$  of (19) which are binary. Thus, we have  $(Z, Y) = \sum_{l=1}^p \lambda_l (Z^l, Y^l)$  for a given nonnegative vector  $\lambda \geq 0$  such that  $\sum_{l=1}^p \lambda_l = 1$ . The operational cost term  $C_{OP}(Y)$  in (21) is additive in  $Y$ , since

$$C_{OP}(Y) = \sum_{t=1}^T \sum_{d=1}^t dY_{t,d} = \sum_{t=1}^T \sum_{d=1}^t d \left( \sum_{l=1}^p \lambda_l Y_{t,d}^l \right) = \sum_{l=1}^p \lambda_l \left( \sum_{t=1}^T \sum_{d=1}^t dY_{t,d}^l \right) = \sum_{l=1}^p \lambda_l C_{OP}(Y^l).$$

So, if  $(Z, Y)$  satisfies for each  $i \in N$  that

$$\gamma_i(Z, Y) = \sum_{l=1}^p \lambda_l \gamma_i(Z^l, Y^l), \quad (22)$$

then the additive relation follows for the penalty cost term  $P(Z, Y)$  in (21), because

$$P(Z, Y) = \sum_{i \in N} \gamma_i(Z, Y) p_i = \sum_{i \in N} \left( \sum_{l=1}^p \lambda_l \gamma_i(Z^l, Y^l) p_i \right) = \sum_{l=1}^p \lambda_l \left( \sum_{i \in N} \gamma_i(Z^l, Y^l) p_i \right) = \sum_{l=1}^p \lambda_l P(Z^l, Y^l),$$

and the total cost is additive in  $(Z, Y)$ , i.e.  $C(Z, Y) = \sum_{l=1}^p \lambda_l C(Z^l, Y^l)$ . So, if condition (22) is true, the optimal cost is a convex combination of binary extreme point costs and it directly implies that there should be an integer extreme point  $l^*$  satisfying  $C(Z^{l^*}, Y^{l^*}) \leq C(Z, Y)$ . This is our desired contradiction.

*Proof of condition (22):* Note that  $\gamma_i(Z, Y) \leq \sum_{l=1}^p \lambda_l \gamma_i(Z^l, Y^l)$  is trivial, since the optimal value of (20) is a convex function of the right-hand-side argument  $(Z, Y)$ . Also, we have that  $\gamma_i(Z^l, Y^l) = 1$  when the operation encoded in  $Y^l$  covers order  $i$ , else it is equal to 0. So, the right-hand-side of (22) yields  $\sum_{l=1}^p \lambda_l \gamma_i(Z^l, Y^l) = 1 - \sum_{l: Y^l \text{ covers } i} \lambda_l$ .

We need to show that the left-hand-side of (22) is also equal to the above value. There is two cases:

1. Suppose that for each  $l \in \{1, \dots, p\}$  with  $0 < \lambda_l < 1$ , the operation encoded in  $Y^l$  covers order  $i \in N$  at most in one dispatch. In case that  $Y^l$  covers  $i$  exactly once, then  $(\lambda_l Y^l, \lambda_l Z^l)$  will add in (20) exactly one type 2 arc  $m_{i,t}^d$  with capacity  $\lambda_l > 0$ , where  $Y_{t,d}^l$  is such that  $d_i \leq d$  and  $\tau_i \geq t$ . Also,  $(\lambda_l Y^l, \lambda_l Z^l)$  will produce a zero cost path from  $\tau_i$  to  $S^i$  with capacity  $\lambda_l$  that uses arc  $m_{i,t}^d$ . On the other hand, if  $Y^l$  does not cover  $i$  there will be no additional paths to  $S^i$ . If we put all these solutions  $Y^l$  together for each  $l \in \{1, \dots, p\}$  with  $0 < \lambda_l < 1$  and form  $Y = \sum_{l=1}^p \lambda_l Y^l$ , the binding cut between  $\tau_i$  and  $S^i$  with zero-cost flows will be defined by  $U = \{1, \dots, \tau_i\}$  with capacity  $\sum_{l: Y^l \text{ covers } i} \lambda_l$ . So, given that the cut is always binding, if we put these paths together in one single network it does not affect the output and  $\gamma_i(Z, Y) = 1 - \sum_{l: Y^l \text{ covers } i} \lambda_l$ . Figure 10 provides an example of this network showing the arc capacities of subproblem (20) for order  $i$ . This case has three integer extreme points  $Y$ :  $Y^1, Y^2$  and  $Y^3$  defining  $Y = \lambda_1 Y^1 + \lambda_2 Y^2 + \lambda_2 Y^3$  and  $1 = \lambda_1 + \lambda_2 + \lambda_2$  for  $\lambda \geq 0$ .  $Y^1$  and  $Y^2$  cover order  $i$ , but  $Y^3$  does not. It is clear that the maximum zero-cost flow from  $\tau_i$  to  $S^i$  is equal to the capacity of the cut  $U$  equal to  $\lambda_1 + \lambda_2 < 1$ . So,  $\gamma_i = 1 - \lambda_1 - \lambda_2$ .
2. A potential problem could occur if an operation covers an order more than once in multiple dispatches. For example, suppose that there exists an operation  $l^1$  with  $0 < \lambda^{l^1} < 1$  such that  $Y^{l^1}$  covers order  $i$  twice and that there exists another operation  $l^2$  not covering  $i$  such that the vehicle is at the depot when operation  $l^1$  dispatches the latest dispatch covering  $i$ . Then, an ‘‘artificial’’ coverage is created for order  $i$ . Figure 11 illustrates this problem. In this example, operation  $l = 1$  with weight  $\lambda_1 = 0.5$  waits at the

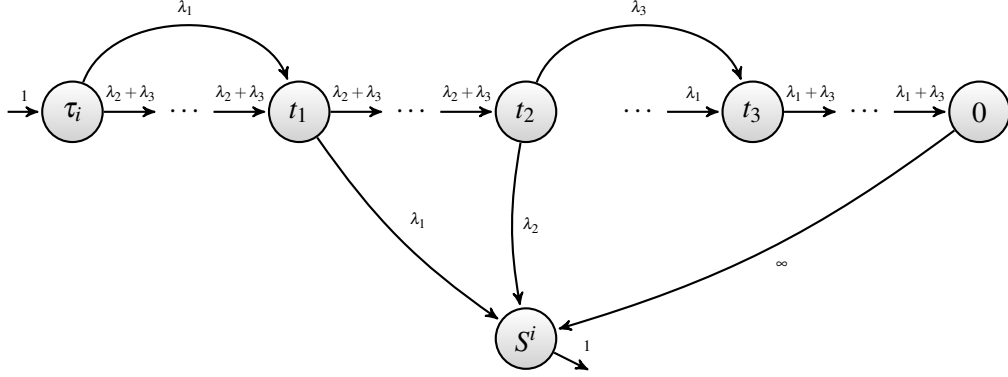


Figure 10: Example of a convex combination of three operations in  $i^{th}$  order subproblem.

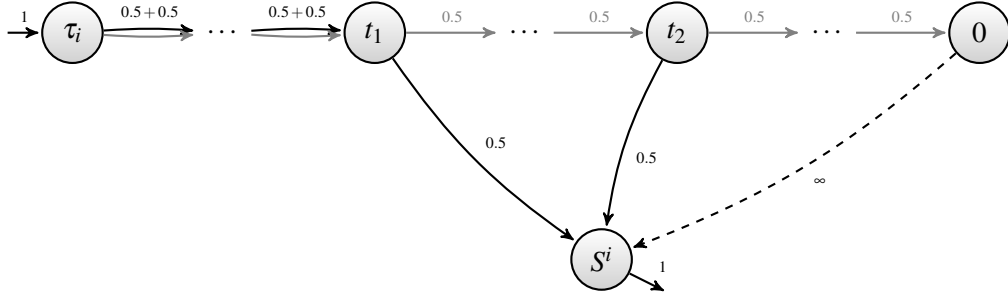


Figure 11: Example of a convex combination of two operations where subproblem for order  $i$  is not additive in the argument  $(Z, Y)$ .

depot until  $t_1$ , covers order  $i$  at  $t_1$ , returns at  $t_2$  and covers order  $i$  again at  $t_2$ . Operation  $l = 2$  with weight  $\lambda_2 = 0.5$  waits at the depot all the time (between  $\tau_i$  and 0). We have that  $0.5\gamma_i(Z^1, Y^1) + 0.5\gamma_i(Z^2, Y^2) = 0.5$ , but  $\gamma_i(0.5(Z^1, Y^1) + 0.5(Z^2, Y^2)) = 1 - 0.5 - \min\{0.5, 0.5\} = 0$ . So condition (22) does not hold. Fortunately, we can prove that there exists an alternative set of operations  $l \in E$  such that  $Y$  can also be written as  $Y = \sum_{l \in E} \lambda_l Y^l$  and such that condition (22) holds.

Let us solve this problem for the example in Figure 11 first. Define  $Y^3$  and  $Y^4$  as follows. Let

$$Y_{t,d}^3 := \begin{cases} Y_{t,d}^1 & t > t_2, 1 \leq d \leq t \\ Y_{t,d}^2 & t \leq t_2, 1 \leq d \leq t \end{cases} \text{ and } Y_{t,d}^4 := \begin{cases} Y_{t,d}^2 & t > t_2, 1 \leq d \leq t \\ Y_{t,d}^1 & t \leq t_2, 1 \leq d \leq t \end{cases}.$$

Note that  $Y = 0.5Y^3 + 0.5Y^4$  and, thus, this new decomposition does not affect operational costs. Also, it covers the same amount of orders plus the ‘‘artificial’’ coverage which is now valid. So  $0.5\gamma_i(Z^3, Y^3) + 0.5\gamma_i(Z^4, Y^4) = \gamma_i(0.5(Z^3, Y^3) + 0.5(Z^4, Y^4)) = 0$ . Figure 12 presents this solution.

The general proof can be constructed by induction on  $r_1 + r_2$ , where  $r_1$  is the total number of additional dispatches covering  $i \in N$  in operations inside  $S$ , and  $r_2$  is the number of operations not covering  $i$  in  $S$  with the vehicle at the depot at a time  $t^*$  where another operation  $l' \in S$  executes a dispatch covering  $i$  which is not the earliest such dispatch.

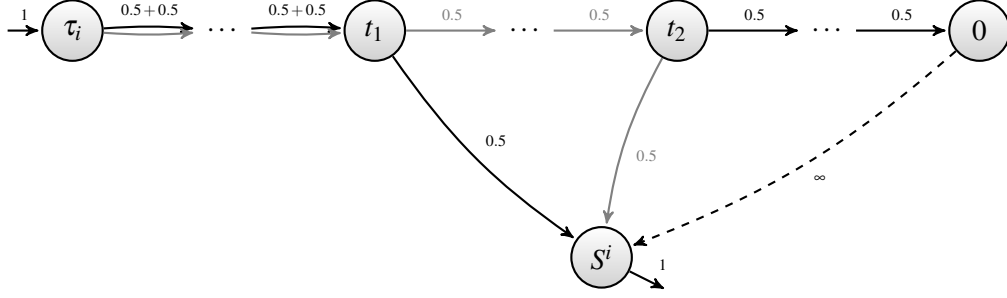


Figure 12: Same example with two operations where subproblem for order  $i$  is additive in the argument  $(Z, Y)$ .

- Case  $r_1 = 0, r_2 = 0$ : This case is trivial, since the set  $S : Y = \sum_{l \in S} \lambda_l Y^l$  satisfies (22).
- Case  $r_1 > 0, r_2 = 0$ : This case is also trivial, since the multiple dispatches cannot be used to generate “artificial coverages” and any  $S$  such that  $Y = \sum_{l \in S} \lambda_l Y^l$  satisfies (22).
- Case  $r_1 = 0, r_2 > 0$ : This case is impossible, by the definition of  $r_2$  ( $r_1 = 0 \implies r_2 = 0$ ).
- Case  $r_1 > 0, r_2 > 0$ : Let  $l^1 \in S$  be the operation with a repeated dispatch to  $i$  at time  $t^*$  such that there exists another operation  $l^2 \in S$  not covering  $i$  and with the vehicle available at the depot at time  $t^*$ . Construct two new operations  $l^3$  and  $l^4$  as follows:

$$Y_{t,d}^{l^3} := \begin{cases} Y_{t,d}^{l^1} & t > t^*, 1 \leq d \leq t \\ Y_{t,d}^{l^2} & t \leq t^*, 1 \leq d \leq t \end{cases} \quad \text{and} \quad Y_{t,d}^{l^4} := \begin{cases} Y_{t,d}^{l^2} & t > t^*, 1 \leq d \leq t \\ Y_{t,d}^{l^1} & t \leq t^*, 1 \leq d \leq t \end{cases}.$$

We have three cases:

- If  $\lambda_{l^1} < \lambda_{l^2}$ , we have  $Y = \sum_{l \in S \setminus \{l^1, l^2\}} \lambda_l Y^l + \lambda_{l^1} (Y^{l^3} + Y^{l^4}) + (\lambda_{l^2} - \lambda_{l^1}) Y^{l^2}$  and  $r_1$  decreases by one. Use induction with  $S' = S \setminus \{l^1\} \cup \{l^3, l^4\}$ .
- If  $\lambda_{l^2} < \lambda_{l^1}$ , we have  $Y = \sum_{l \in S \setminus \{l^1, l^2\}} \lambda_l Y^l + \lambda_{l^2} (Y^{l^3} + Y^{l^4}) + (\lambda_{l^1} - \lambda_{l^2}) Y^{l^1}$  and  $r_2$  decreases by one. Use induction with  $S' = S \setminus \{l^2\} \cup \{l^3, l^4\}$ .
- If  $\lambda_{l^2} = \lambda_{l^1}$ , set  $Y = \sum_{l \in S \setminus \{l^1, l^2\}} \lambda_l Y^l + \lambda_{l^1} (Y^{l^3} + Y^{l^4})$  and  $r_1$  and  $r_2$  each decrease by one. Use induction with  $S' = S \setminus \{l^1, l^2\} \cup \{l^3, l^4\}$ .

□

## 7.4 ALP solution pruning

We can reduce the computational effort involved in getting the ALP optimal policy defined by (9) with the following proposition:

**Proposition 7.1** (ALP solution pruning). *Suppose  $\delta \in \mathcal{A}_R^t$  is a feasible dispatch distance at state  $(t, R, P)$  and its related ALP solution to (9) is  $\{\mathbf{a}(\delta), \mathbf{b}(\delta), \mathbf{v}(\delta)\}$ . Let  $\mu \in \mathcal{A}_R^t$  be a different feasible dispatch distance. If*

$$\begin{aligned} & \alpha \delta + \sum_{i \in R_\delta} a_i(\delta)^{t-\delta} + \sum_{i \in P} (g_{ii}^\delta a_i(\delta)^{t-\delta} + \bar{g}_{ii}^\delta b_i(\delta)^{t-\delta}) - \sum_{k=1}^{t-\delta} v_k(\delta) \\ & < \alpha \mu + \sum_{i \in R_\mu} a_i(\delta)^{t-\mu} + \sum_{i \in P} (g_{ii}^\mu a_i(\delta)^{t-\mu} + \bar{g}_{ii}^\mu b_i(\delta)^{t-\mu}) - \sum_{k=1}^{t-\mu} v_k(\delta), \end{aligned}$$

then  $\mu$  is suboptimal for (9) and can be discarded before solving its related ALP.

*Proof.* The proof is based on the fact that  $\{\mathbf{a}(\delta), \mathbf{b}(\delta), \mathbf{v}(\delta)\}$  is also a feasible solution for the ALP problem related to  $\mu$ . By proposition (7.1) and the feasibility of  $\mathbf{a}(\delta), \mathbf{b}(\delta), \mathbf{v}(\delta)$  in any ALP problem we get

$$\begin{aligned}
& \alpha\delta + \sum_{i \in R_\delta} a_i(\delta)^{t-\delta} + \sum_{i \in P} \left( g_{ii}^\delta a_i(\delta)^{t-\delta} + \bar{g}_{ii}^\delta b_i(\delta)^{t-\delta} \right) - \sum_{k=1}^{t-\delta} v_k(\delta) \\
& < \alpha\mu + \sum_{i \in R_\mu} a_i(\delta)^{t-\mu} + \sum_{i \in P} \left( g_{ii}^\mu a_i(\delta)^{t-\mu} + \bar{g}_{ii}^\mu b_i(\delta)^{t-\mu} \right) - \sum_{k=1}^{t-\mu} v_k(\delta) \\
& \leq \alpha\mu + \max_{\{(\mathbf{a}, \mathbf{b}, \mathbf{v}) \in (7b)-7i\}} \sum_{i \in R_\mu} a_i^{t-\mu} + \sum_{i \in P} \left( g_{ii}^d a_i^{t-\mu} + \bar{g}_{ii}^d b_i^{t-\mu} \right) - \sum_{k=1}^{t-\mu} v_k,
\end{aligned}$$

and this proves that the dispatch distance  $\delta$  yields a lower approximate expected cost than  $\mu$  for the ALP policy.  $\square$