An Approximate Dynamic Programming Approach to Urban Freight Distribution with Batch Arrivals

Wouter van Heeswijk(✉), Martijn Mes, and Marco Schutten

Department of Industrial Engineering and Business Information Systems,
University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
{w.j.a.vanheeswijk,m.r.k.mes,m.schutten}@utwente.nl

Abstract. We study an extension of the delivery dispatching problem (DDP) with time windows, applied on LTL orders arriving at an urban consolidation center. Order properties (e.g., destination, size, dispatch window) may be highly varying, and directly distributing an incoming order batch may yield high costs. Instead, the hub operator may wait to consolidate with future arrivals. A consolidation policy is required to decide which orders to ship and which orders to hold. We model the dispatching problem as a Markov decision problem. Dynamic Programming (DP) is applied to solve toy-sized instances to optimality. For larger instances, we propose an Approximate Dynamic Programming (ADP) approach. Through numerical experiments, we show that ADP closely approximates the optimal values for small instances, and outperforms two myopic benchmark policies for larger instances. We contribute to literature by (i) formulating a DDP with dispatch windows and (ii) proposing an approach to solve this DDP.

Keywords: Urban distribution · Transportation planning · Consolidation · Approximate dynamic programming

1 Introduction

In the field of urban freight logistics, the need for consolidation centers at the edge of urban areas is becoming increasingly important [11]. Due to the external costs of freight transport – such as congestion, air pollution, and noise hindrance – more efficient goods transport within the city center is necessary. Governments seek to reduce the negative influence of large trucks in urban areas. Imminent regulations are, e.g., restricted access areas, and road pricing for heavy vehicles. Such developments spur the need for transshipments at the edge of urban areas. Transshipments allow both for bundling goods – such that vehicle capacity can be used more efficiently – and dispatching environment-friendly vehicles such as electric vans on the last mile. However, the introduction of an additional transshipment in the supply chain also poses new challenges. The challenge we study is inspired by a project on urban freight logistics, in which various logistics partners participate.

F. Corman et al. (Eds.): ICCL 2015, LNCS 9335, pp. 61–75, 2015.
DOI: 10.1007/978-3-319-24264-4_5

We adopt the perspective of the party in charge of the consolidation center, to which we refer as the 'hub operator'. We focus on the timing of dispatching orders for delivery, in an environment where the batch arrivals of goods at the hub are not fully controlled by the hub operator. Hub operators are generally small parties that deploy rules-of-thumb for dispatching orders. They have a certain degree of knowledge of order arrivals over a relatively short horizon; order arrivals are usually announced at most 24 hours in advance. The operators also have expectations regarding future order arrivals – e.g., based on historical data – which can be modeled as stochastic arrivals.

We consider an extension of the Delivery Dispatching Problem (DDP) that includes dispatch windows as an order characteristic. This extension is particularly relevant in an urban distribution context, where deliveries within specified time windows are the norm. As opposed to traditional DDPs, we consider a finite planning horizon, allowing to capture time-dependent arrival processes (e.g., holiday weeks). Commonly studied shipment consolidation policies fall short to aid decision-making in this context. It does not suffice to determine when to dispatch the orders in inventory, one also needs to determine which subset of orders to dispatch. In our DDP, order batches are dynamically revealed to the hub operator. Some orders may arrive at the consolidation center without advance notice, others orders may be scheduled to arrive at a future point in time. After orders have physically arrived at the consolidation center, they can be dispatched to the customers in the city. An arriving batch may contain orders with dispersed destinations, various dispatch windows, distinct load sizes, etc. Directly distributing an arriving batch may therefore render poor solutions. Instead, waiting for future batches to arrive could yield order clusters for which better solutions are available. This entails waiting for known batches, but also the inclusion of future orders that may have uncertain properties. Based on both the available knowledge regarding current orders and anticipation of new orders, the operator is able to make informed waiting decisions.

Due to the dynamic and stochastic nature of our DDP – combined with the large amount of states – we must deal with various computational challenges. We consider our study to be part of a two-phase solution approach. In the first phase – which is the scope of this paper – the hub operator decides which orders to dispatch at the current decision moment. The dispatch decision is based on known information and probabilistic arrivals; estimates for the direct costs and downstream costs are used. In the second phase, the operator solves a detailed VRP for the selected set of orders. With this paper, we aim to contribute to existing literature with (i) the formulation of a Markov model for DDPs with dispatch windows, and (ii) an approach to provide high-quality solutions for larger-sized DDPs.

2 Literature Review

In this section, we analyze the existing literature on the DDP and related topics. We refer to recent literature studies for overviews of these problems, and highlight several studies that address problems comparable to ours. We address various solution approaches, and discuss their suitability for our problem type. Finally, we point out the literature gap that we aim to address.

Optimization problems that are both stochastic and dynamic are notoriously hard to solve [10]. The use of stochastic information in transportation problems is recognized as an important aspect of optimization, yet its incorporation in solution methods is still an ongoing development [10,14]. Mathematical programming and (meta)heuristics have traditionally been applied to handle high-dimensional problems in transportation. However, these methods generally do not cope well with stochastic information being revealed over time [10]. Suitable solution approaches tend to be either based on stochastic modeling or scenario sampling [5,8]; the latter is generally applied to fit heuristics and mathematical programs towards stochastic problems. Incorrect sampling may misrepresent the stochastic process [5]. Stochastic models represent all possible outcomes, and therefore in general require more computational effort. As a result, they are better fit for preprocessed decisions than for online decision-making.

We classify our problem as a DDP. In a DDP, orders arrive according to a stochastic process and are dispatched in batches [6]. Solving the DDP yields a shipment consolidation policy, indicating when to dispatch orders held in inventory. We briefly described the DDP in the introduction. For a more elaborate definition we refer to Minkoff [6], who states that all eligible routes are predefined input for DDPs. The performance of shipment consolidation policies is generally evaluated based on efficiency (vehicle capacity utilization) and/or timeliness (time between order arrival and dispatch). Policies are either recurrent (i.e., dependent on the state of the problem) or non-recurrent [9]; we study a DDP with a recurrent policy. The stochastic and dynamic nature of such DDPs gives rise to Markov decision problems [6]. Although a Markov decision model is a useful framework to describe decision problems with dynamic and stochastic elements, practical implementations generally suffer from intractably large state spaces and expected values that cannot be calculated exactly [6,8,9]. Relatively little work has been done on optimizing consolidation policies; the majority of DDP literature focuses on testing the performance of existing policies [1,7]. Most DDP literature only considers weight and arrival time as order properties, while results are valid for a limited set of distributions. A more generic approach based on a batch Markovian arrival process is presented by Bookbinder *et al.* [1]; allowing for arrival properties that follow any distribution. Although able to cope with a variety of arrival processes, enumeration of the transition matrix is still required. Even when applying techniques to simplify this procedure, complete enumeration is computationally challenging to describe batch arrivals consisting of order types with multiple stochastic properties such as dispatch windows.

Although the Vehicle Routing Problem (VRP) addresses routing decisions rather than dispatching, some characteristics are shared. Unlike the DDP, the inclusion of time windows is a common property of VRPs. Ritzinger *et al.* [12] provide an overview of dynamic and stochastic VRP literature, which generally considers re-optimization during the execution of routes. A particularly relevant class they describe considers dynamic order requests combined with stochastic customer information. For this class, they distinguish between preprocessed decision support (the sub-class we study) and online decision making. For preprocessed decision support, a number of stochastic modeling approaches is mentioned. Solutions for online decision problems tend to focus more on sampling.

Another subject related to the DDP is the Inventory Routing Problem (IRP) [6]. The IRP is concerned with repeated stock replenishment from a facility to a fixed set of delivery locations, the decision when to visit a location, and the quantity of product to be delivered are the decision of the facility. Each location consumes the product at a given – possibly stochastic – rate. As such, the IRP also considers a dispatching decision. However, deliveries in an IRP are not order-based; goods can be dispatched to any customer at any decision moment. Furthermore, one or several types of goods have to be distributed along multiple customers. Coelho *et al.* [3] provides a recent overview of IRP literature. They state that for the solution of stochastic IRPs, generally either Markov models are solved in a heuristic manner, or mathematical programming is applied. For IRPs with both dynamic and stochastic properties they mention only few works. Coelho *et al.* [2] solve a problem in this IRP class heuristically, forecasting a single scenario based on exponential smoothing of historical data.

Finally, we refer to several Service Network Design (SND) studies mentioned in SteadiSeifi *et al.* [14]. SND is concerned with the selection and timing of transportation services. Known solution approaches make use of mathematical programming, (meta)heuristics, and graph theory. Most SND studies focus on deterministic instances. Lium *et al.* [5] propose a stochastic extension to their mathematical program, adding scenarios to reflect uncertain future demand. The authors state that generating a compact yet representative scenario tree is one of the key challenges in this approach.

We did not encounter existing DDP literature that mathematically formulates a dispatch problem for orders with time windows. We aim to contribute to literature by formulating our DDP as a Markov decision process that captures both the stochastic and dynamic nature of the order arrival process. Dynamic programming (DP) can be used to solve such models to optimality, but instance sizes quickly grow too large for exact solutions. Topaloglu and Powell [15] present a stochastic modeling framework for solving dynamic resource-allocation problems, proposing the application of approximate dynamic programming (ADP). Various successful ADP applications can be found in transportation literature [8,14]. Following frameworks such as [9,10,15], we develop an ADP approach to solve our DDP with dispatch windows.

3 Problem Formulation

This section introduces the planning problem. We describe the problem in a generic way, making it applicable to a variety of instances. We assume that the characteristics of arriving orders are stochastic and have a known associated probability distribution. Our problem formulation allows to include both deterministic and stochastic future orders. We consider a finite planning horizon, during which batches of orders can arrive at the consolidation center. Dispatching decisions are made at fixed decision moments within the planning horizon, with constant time intervals separating the decision moments. We model the arrival rates of order batches, the number of orders in a batch, order sizes, order destinations, and dispatch times as stochastic variables. When a batch of orders arrives at the center, the exact properties of the orders are revealed.

The decision problem that we address is the choice of which orders to dispatch at the current decision moment. To make an informed decision, we require insight in the effects of postponing the dispatch of orders. Postponed orders may be combined with future orders against lower costs than when dispatched at the current decision moment. We therefore consider optimization over a planning horizon; orders not known at the current decision moment are probabilistic. We assess all possible realizations of stochastic order arrivals, and plan these arrivals as if they were actual orders. For both the deterministic and stochastic orders belonging to a given realization, we compute the costs of dispatching. The costs of dispatching stochastic orders are required to quantify the expected costs. By incorporating stochastic order arrivals, we can compute the expected costs of the various dispatch decisions for the currently known orders.

Consider an urban area with a fixed set of potential order destinations, which are delivered via a consolidation center at the edge of the area. Our representation of the urban distribution network is as follows. Let $\mathcal{G} = \{\mathcal{V}, \mathcal{A}\}$ be a directed and complete graph with $\mathcal{V}$ being the set of vertices and $\mathcal{A}$ being the set of arcs. $\{0\} \in \mathcal{V}$ represents the consolidation center in the network. The remaining vertices signify the subset of order destinations $\mathcal{V}' = \{1, 2, \ldots, |\mathcal{V}'|\}$, with $\mathcal{V}' = \mathcal{V} \setminus \{0\}$. The distances between any pair of vertices in the graph are known.

We consider a planning horizon that contains decision moments with fixed intermediate time intervals. Let $\mathcal{T} = \{0, 1, \ldots, T\}$ be the set containing all decision moments, and $t \in \mathcal{T}$ describe any decision moment within the planning horizon. We consider a homogeneous fleet (i.e., a set of identical vehicles), although our method is able to cope with heterogeneous fleets as well. We distinguish between sets of primary vehicles $\mathcal{Q}^{pr}$ and secondary vehicles $\mathcal{Q}^{se}$. We assume that the secondary fleet has an infinite size, and is either an actual transport alternative (e.g., renting an additional vehicle in case of shortage) or a dummy fleet with infinite costs. A dummy fleet serves as bound on capacity, without having to explicitly calculate the capacity constraints for each decision. We only assign vehicles in $\mathcal{Q}^{se}$ if no more vehicles in $\mathcal{Q}^{pr}$ are available. We assume that dispatching a secondary vehicle is always more expensive than dispatching a primary vehicle. To ease the presentation, we assume that every dispatched vehicle has a fixed route duration of $\tau_{route} \geq 1$ (this assumption can be easily relaxed). When dispatching at t, the vehicle will be available again at $t + \tau_{route}$. For decision-making purposes, we keep track of the availability of primary vehicles now and in the future. Let $r \in [0, \tau_{route} - 1]$ be the number of time intervals before a dispatched vehicle returns. Because all vehicles are available again at $t + \tau_{route}$, we only keep track of availability up to $t + \tau_{route} - 1$. Let $q_{t,r}$ denote the number of primary vehicles available for dispatch at $t+r$. It follows that $q_{t,0}$ vehicles are available for dispatch at t. We record primary fleet availability in the vector $Q_t = (q_{t,0}, q_{t,1}, \dots, q_{t,t+\tau_{route}-1})$.

Every order is characterized by four properties: destination, load size, earliest dispatch time, and latest dispatch time. The order destination (i.e., the customer location) is represented by a vertex $v \in \mathcal{V}'$. Let $\mathcal{L} = \{\frac{1}{k}, \frac{2}{k}, \dots, 1\}$ be the discretized set of viable load sizes, with integer $k \geq 1$, and 1 representing a full load for an urban vehicle. The size of an order is given by $l \in \mathcal{L}$. The hard dispatch window of an order is given by earliest dispatch time $e \in \mathcal{E}$ and latest dispatch time $d \in \mathcal{D}$. Both indices are relative to the decision moment t; at the decision moment $t+1$ all indices of orders in inventory are reduced by 1. Only orders with $e = 0$ can be dispatched. Order types with $e > 0$ describe pre-announced future orders, that will be delivered to the hub at time $t+e$. We define a maximum length of the dispatch window τ_{window}, such that $d \in [e, e+\tau_{window}]$. Every unique combination of the four properties represents an order type. Let $I_{t,v,l,e,d} \in \mathbb{Z}_+$ be the number of a given order type in inventory at t. We denote the information regarding all known orders at t as $I_t = (I_{t,v,l,e,d})_{v \in \mathcal{V}', l \in \mathcal{L}, e \in \mathcal{E}, d \in \mathcal{D}}$. The state of the system at t, $S_t \in \mathcal{S}$, combines primary fleet availability with available orders, and is represented by

$$S_t = (Q_t, I_t)_{\forall v \in \mathcal{V}', l \in \mathcal{L}, e \in \mathcal{E}, d \in \mathcal{D}} \quad , \forall t \in \mathcal{T} . \tag{1}$$

For $t \geq 1$, let $\mathcal{O}_t = \{0, 1, \dots, o_t^{max}\}$ be the set containing the number of possible order arrivals between decision moments $t-1$ and t. Let $o_t \in \mathcal{O}_t$ be a realization of the number of orders arriving between $t-1$ and t. Furthermore, we set $l^{max} \in \mathbb{Z}_+$ as the maximum number of orders that can be held in inventory, i.e., the maximum inventory remaining after a decision.

For every decision moment t in the planning horizon, we decide which orders in inventory to dispatch. Orders that are not dispatched remain in inventory, and are available at the next decision moment. Let the integer variable $x_{t,v,l,e,d}$ describe the number of a specific order type to be dispatched at t. A feasible action at decision moment t is given by

$$x_t(S_t) = (x_{t,v,l,e,d})_{\forall v \in \mathcal{V}', l \in \mathcal{L}, e \in \mathcal{E}, d \in \mathcal{D}}, \tag{2}$$

where

$$\sum_{v \in \mathcal{V}', l \in \mathcal{L}, e \in \mathcal{E}, d \in \mathcal{D}} (I_{t,v,l,e,d} - x_{t,v,l,e,d}) \leq l^{max}, \tag{3}$$

$$x_{t,v,l,e,d} \leq I_{t,v,l,e,d} \quad , \forall v \in \mathcal{V}', \forall l \in \mathcal{L}, \forall e \in \mathcal{E}, \forall d \in \mathcal{D} \quad , \tag{4}$$

$$x_{t,v,l,e,0} = I_{t,v,l,e,0} \quad , \forall v \in \mathcal{V}', \forall l \in \mathcal{L}, \forall e \in \mathcal{E} \quad , \tag{5}$$

$$x_{t,v,l,e,d} = 0 \quad , e > 0 \quad , \forall v \in \mathcal{V}', \forall l \in \mathcal{L}, \forall d \in \mathcal{D} \quad , \tag{6}$$

$$x_{t,v,l,e,d} \in \mathbb{Z}_+ \quad , \forall v \in \mathcal{V}', \forall l \in \mathcal{L}, \forall e \in \mathcal{E}, \forall d \in \mathcal{D} \quad . \tag{7}$$

Constraint (3) ensures that after dispatching, no more than the maximum inventory remains at the consolidation center. According to Constraint (4), it is not possible to dispatch more orders of a certain type than available at the decision moment t. Constraint (5) states that all orders that have reached their latest dispatch time must be dispatched. Constraint (6) prevents orders with an earliest dispatch time in the future from being dispatched. Constraint (7) states that only nonnegative integer amounts of orders can be dispatched. The set of feasible actions in a given state is described by $\mathcal{X}_t(S_t)$.

4 Markov Model

We model the operator's decision problem as a Markov model. This model considers all possible realizations of orders arrivals during the planning horizon. With this knowledge, we can make the optimal dispatch decision for the current decision moment. In realistic instances, the state space, action space, and outcome space for such a model will be intractably large. Exactly solving the Markov model is therefore not possible within a reasonable time. In Section 6, we solve some toy-sized instances of the Markov model using DP. The ADP approach as outlined in Section 5 is applied to larger instances.

Every action $x_t(S_t)$ has associated direct costs $C(S_t, x_t)$. The direct costs are the sum of fixed dispatching costs per vehicle, variable transportation costs, and handling costs. As the focus of this paper is on the consolidation policy, we do not explicitly consider routing. Instead, we use the classic approximation of Daganzo [4] to estimate the transportation costs for a dispatched set of orders. This approximation is known to provide good estimates of total route distances [13], given constraints on vehicle capacity, number of destinations, and shape of the service area. These constraints are likely to be fulfilled in an urban distribution setting. The approximation is based on the average distances between the depot and the customers, the number of customer locations visited, the size of the service area, and the capacity of the vehicles. We consider fixed handling costs per visited customer; note that this provides an incentive to simultaneously deliver multiple orders to a customer.

To model the uncertainties with respect to the properties of arriving orders, we introduce six stochastic variables. These are (i) the number of orders arriving O_t, (ii) the destination V, (iii) the order size L, (iv) the earliest dispatch time E, (v) the length of the dispatch window D_{window}, and (vi) the latest dispatch time $D = E + D_{window}$. The corresponding probability distributions are discrete and finite. To capture all probability distributions into a single variable, we define the exogenous information variable $\tilde{I}_{t,v,l,e,d} \in \mathbb{Z}_+, t \geq 1$, which indicates the number of arrivals of a specific order type. Furthermore, we introduce a generic variable W_t that describes all exogenous information, i.e., all orders arriving between $t-1$ and t:

$$W_t = [\tilde{I}_{t,v,l,e,d}]_{\forall v \in \mathcal{V}', \forall l \in \mathcal{L}, \forall e \in \mathcal{E}, \forall d \in \mathcal{D}} \ , t \geq 1. \tag{8}$$

There exists a finite number of realizations of W_t. Let Ω_t be the set of possible batch arrivals between $t-1$ and t, and $\omega_t \in \Omega_t$ be a realization of the random variables occurring with $P(W_t = \omega_t)$.

We proceed to describe the transition from a state S_t to the next state S_{t+1}. The transition is affected by the action x_t and the new arrivals W_{t+1}. We first describe the effects of x_t. Orders not dispatched at t remain in inventory, hence must be included in S_{t+1}. As indices e and d are adjusted over time, we introduce two new variables to properly process the conversion. Let $e' = \max\{0, e-1\}$ and $d' = d-1$. Since $e < 0$ does not affect our decision making, capping e' at 0 reduces the number of possible order types. Let $\bar{q}_t \in \{0, \dots, |\mathcal{Q}^{pr}|\}$ be the number of primary vehicles dispatched at t; combined with Q_t this information suffices to compute Q_{t+1}. We represent new arrivals with the information variable W_{t+1}. This gives us the transition function

$$S_{t+1} = S^M(S_t, x_t, W_{t+1}), \tag{9}$$

where

$$I_{t+1,v,l,e',d'} = I_{t,v,l,e,d} - x_{t,v,l,e,d} + \tilde{I}_{t+1,v,l,e',d'}, \qquad \forall t \in \mathcal{T}, \forall v \in \mathcal{V}', \forall l \in \mathcal{L}, \forall e \in \mathcal{E}, \forall d \in \mathcal{D}, \tag{10}$$

$$q_{t+1,r} = \begin{cases} q_{t,r+1} - \bar{q}_t & \text{if } r < \tau_{route} - 1 \ , \\ |\mathcal{Q}^{pr}| & \text{if } r = \tau_{route} - 1 \ , \end{cases} \forall r \in [0, \tau_{route} - 1]. \tag{11}$$

Constraint (10) states that for every order type, we have the amount of the order type in state S_t, minus the amount of the order type that was dispatched, plus the amount of the order type that arrived between t and $t+1$. Constraint (11) ensures that Q_{t+1} is consistently updated. Having described the transition function, we now introduce the optimality equation that must be solved:

$$V_t(S_t) = \min_{x_t \in \mathcal{X}_t(S_t)} \left(C(S_t, x_t) + \sum_{\omega_{t+1} \in \Omega_{t+1}} P(W_{t+1} = \omega_{t+1}) V_{t+1}(S_{t+1} | S_t, x_t, \omega_{t+1}) \right). \tag{12}$$

We proceed to describe the state space. Between every two consecutive decision moments $t-1$ and t, we can have $o_t \in \{0, \dots, |\mathcal{O}_t| - 1\}$ new orders arriving. Every arriving order can have any of the unique order types, given the constraints on the dispatch window. Before the new arrivals occur, we can have up to l^{max} orders in inventory. Hence, we can have at most $l^{max} + |\mathcal{O}_t| - 1$ orders at a given decision moment. A state can be any feasible combination of order types available at t, combined with any vector Q_t.

Next, we describe the action space. At every decision moment, we decide which orders to dispatch. Every combination of orders to dispatch represents a unique action. Orders that are not dispatched remain in inventory, and may be dispatched at the next decision moment. As we do not consider routing options, a unique selection of orders to dispatch equals exactly one action.

The transition from one state to another is determined by the current state, the used action, and the realization of the random variables. The remaining inventory before new orders arrive is deterministic. The probability of o_t orders arriving is given by $P(O_t = o_t)$. The probability of an arriving order being of a certain order type is given by the multivariate distribution $P(V, L, E, D)$. V, L and E are independent random variables, while D is the sum of the realizations of E and D_{window}.

The outcome space is dependent on the state S_t, the action x_t, and the realization of new arrivals ω_{t+1}. Orders not shipped at decision moment t are with certainty included in S_{t+1}. As route duration is deterministic, so is the change in fleet availability. Therefore, only the order arrivals account for stochasticity. To account for the multiple permutations corresponding to o_t order arrivals, we multiply the probability of o_t orders arriving with a multinomial coefficient. We obtain the following probability function for new arrivals:

$$\begin{aligned} P(W_t = \omega_t) = P(O_t = o_t) & \frac{o_t!}{\prod\limits_{\tilde{I}_{t,v,l,e,d} \in \omega_t} \tilde{I}_{t,v,l,e,d}!} \\ & \prod_{v \in \mathcal{V}', l \in \mathcal{L}, e \in \mathcal{E}, d \in \mathcal{D}} P(V = v, L = l, E = e, D = d)^{\tilde{I}_{t,v,l,e,d}} \quad , t \geq 1. \end{aligned} \tag{13}$$

5 Solution Approach

Realistic-sized problems are intractably large for DP. We resolve computational problems with the state- and outcome space with our ADP approach, while partially addressing the dimensionality of the action space in this paper. We retain the full level of detail in the state description, without enumerating the full state space. By means of Monte Carlo simulation, we approximate the exact values of the DP method [9]. In our ADP implementation, we use the concept of the post-decision state [9]. The post-decision state S_t^x is the state immediately after action x_t, but before the arrival of new information ω_{t+1}. Given our action x_t, we have a deterministic transition from S_t to the so-called post-decision state S_t^x. We express this transition in the function

$$S_t^x = S^{M,x}(S_t), \tag{14}$$

where

$$I_{t,v,l,e,d} = I_{t,v,l,e,d} - x_{t,v,l,e,d}, \tag{15}$$

$$\forall t \in \mathcal{T}, \forall v \in \mathcal{V}', \forall l \in \mathcal{L}, \forall e \in \mathcal{E}, \forall d \in \mathcal{D},$$

$$q_{t,r} = q_{t,r} - \bar{q}_t \quad , \forall r \in [0, \tau_{route} - 1]. \tag{16}$$

The post-decision state has a corresponding value function

$$V_t(S_t^x) = \mathbb{E}\{V_{t+1}(S_{t+1}) | S_t^x\}. \tag{17}$$

Adopting the concept of the post-decision state allows us to represent our problem as a deterministic minimization problem. Although this reduces the computational effort, Equation (17) still requires to evaluate all states in the outcome space. We therefore replace this value function with a single value function approximation $\bar{V}_t^{n-1}(S_t^x)$. n is an iteration counter, representing that we use an estimate from iteration $n-1$ at iteration n. At every decision moment, we take the best action given our value function approximation. Incoming arrivals are generated according to Equation (13). Utilizing the post-decision state and value function approximation for future costs, we solve Equation (18) to minimize the value $\hat{v}_t^n$:

$$\hat{v}_t^n = \min_{x_t \in \mathcal{X}_t(S_t)} (C_t(S_t, x_t) + \bar{V}_t^{n-1}(S_t^x)). \tag{18}$$

Once we obtain our estimate $\hat{v}_t^n$, we can update $\bar{V}_{t-1}^{n-1}(S_{t-1}^x)$. For this, we use the following function:

$$\bar{V}_{t-1}^n(S_{t-1}^x) \leftarrow U^V(\bar{V}_{t-1}^{n-1}(S_{t-1}^x), S_{t-1}^x, \hat{v}_t^n). \tag{19}$$

Table 1 provides an outline of our ADP algorithm.

Table 1. ADP algorithm with post-decision states

```
Step 0 Initialize
       Step 0a:      Initialize $\bar{V}_t^0(S_t)$, $\forall t \in \mathcal{T}$, $\forall S_t \in \mathcal{S}$
       Step 0b:      Set iteration counter to $n = 1$, and set
                     the maximum number of iterations to $N$.
       Step 0c:      Select an initial state $S_0$.
Step 1 For $t = 0$ to $T$ do:
       Step 1a:      Find the best action $\tilde{x}_t^n$ by solving Equation (18).
       Step 1b:      If $t > 0$, then update $\bar{V}_t^{n-1}(S_t)$ using Equation (19).
       Step 1c:      Obtain the post-decision state $S_t^x$ via Equation (14).
       Step 1d:      Obtain a sample realization $W_{t+1}$, calculate $S_{t+1}$
                     with Equation (9)
Step 2 Set $n := n + 1$.
       If $n \leq N$, then go to Step 1.
Step 3 Return $\bar{V}_t^N(S_t^x), \forall t \in \mathcal{T}$.
```

We briefly discuss two options for the function U^V to update $\bar{V}_t^n(S_{t-1}^x)$: lookup and value function approximation (VFA). With the lookup approach, we store an estimate $\bar{V}_t^n(S_t^x)$ for every post-decision state, which is updated based on our observation at the next decision moment. We can speed up this procedure by first completing a full iteration, and then update all post-decision values at once (a procedure known as double pass, see [9]). Although the lookup ADP resolves several computational challenges of dynamic programming, we still need to visit a state to learn about its value. Instead, we want to learn about the value of many states with a single observation. To achieve this, we make use of VFA with the so-called basis function approach, see [9]. Let $\mathcal{F}$ be a set of features, with $f \in \mathcal{F}$ being some variable that partially explains the costs of being in a state. Relevant features for our dispatch problem are, e.g., the total volume of orders in inventory, the number of orders with $d = 0$, and the number of distinct destinations. Let $\phi_f(S_t^x)$ be a basis function of feature f – for example, a cross-product or a polynomial of f – that returns a certain value given S_t^x. Let θ_f^n be a weight corresponding to feature f. Our value function approximation becomes

$$\bar{V}_t^n(S_t^x) = \sum_{f \in \mathcal{F}} \theta_f^n \phi_f(S_t^x), \forall t \in \mathcal{T}. \tag{20}$$

Following [9], the weights θ_f^n are updated using recursive least squares for nonstationary data. Using this procedure, we are able to learn about the value of many states by sampling just a single state. Using VFA, it is therefore not necessary to visit all states in the state space to learn about their value, allowing to handle large state spaces. The key difficulty with VFA is to define basis functions that closely approximate the exact values of states. Good insight in the structure of the problem is required to select features that allow to accurately approximate the true values.

After learning the appropriate weights by completing the procedure in Table 1, VFA can be applied for practical decision making. By calculating the values for the post-decision states corresponding to our initial state, we are able to obtain the best action given the estimate. Only the features of the states, the basis functions, and the corresponding weights are necessary for decision making.

6 Numerical Experiments

First, we solve a toy-sized instance with dynamic programming. We show how both the lookup approach and the VFA approach approximate the exact DP values. Next, we consider larger problems. As these instances are too large to solve exactly, we cannot show convergence results for these. For all instances, we compare the ADP-based simulation results to the results of two myopic benchmark policies, showing how the inclusion of future information impacts decision quality. The first benchmark policy ('Postpone') we deploy in this paper is given in Table 2. It aims to postpone as many orders as possible, until a suitable consolidation opportunity arises or the latest dispatch time is reached. The second benchmark policy ('DirectShipment') always ships orders as soon as possible, as long as primary vehicle capacity is available. 'DirectShipment' sorts and assigns orders just as 'Postpone' describes, and also dispatches secondary vehicles only when necessary. We found that in practice, consolidation policies of comparable complexity are applied by hub operators, followed by manual fine-tuning.

We first describe the properties of our toy problem. We consider a fleet of two primary vehicles; secondary vehicles are twice as expensive as primary vehicles. We consider three distinct customer locations, a random order size from $\{0.2, 0.4, 0.6, 0.8, 1\}$, a maximum inventory of two orders, and a maximum of two arrivals per decision moment. We fix the tour length at $\tau_{route} = 1$. We set $e = 0$ for all orders, and select d from $\{0, 1\}$. All probability distributions are

Table 2. Benchmark policy – Postpone

Step 0	Sort orders.	
	Step 0a:	Sort available orders based on lowest d.
	Step 0b:	Sort available orders with same d based on smallest size.
Step 1	While orders with $d = 0$ are unassigned do:	Assign order with $d = 0$ to vehicle.
Step 2	While remaining inventory exceeds l^{max} do:	Assign first order on list to vehicle.
Step 3	While capacity from already dispatched vehicles remains do:	Assign first order on list to vehicle.

uniform. We define a planning horizon with five decision moments. Although an extremely small problem, the state space already contains about 140,000 states.

The features we use for our VFA are (i) a constant, (ii) the number of vehicles available at the decision moment, (iii) the number of distinct order destinations, (iv) the total volume of orders in inventory, and (v) the square of the volume of orders in inventory. In Figure 1 and Figure 2, we show for two initial states (one without initial inventory, the other with four orders at the decision moment) how both the lookup approach and the VFA approach converge to the optimal values found with DP. In the first number of iterations, the estimates fluctuate due to the inability to accurately compute expected costs. However, by learning the values of visited states, ADP starts recognizing good actions.

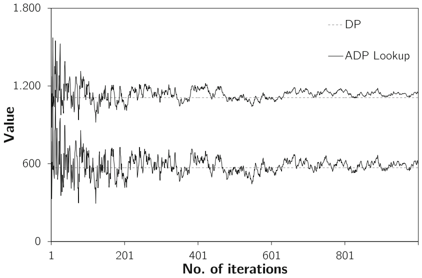

Fig. 1. Approximation of exact value with Lookup

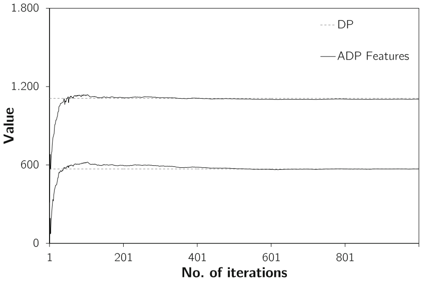

Fig. 2. Approximation of exact value with VFA

From here on, we focus only on ADP with VFA using basis functions. By completing the algorithm in Table 1, we obtain a set of weights for every decision moment. When in a given state, with these weights we estimate the values of all reachable post-decision states. Hence, ADP results in a policy, which we use to solve a deterministic decision problem. We apply the learned policy in a Monte Carlo simulation on a variety of initial states, comparing its performance

Table 3. Comparison between ADP and benchmark policies for small instances

	Average costs	Standard deviation	Average deviation from optimal	Lowest deviation from optimal	Highest deviation from optimal
DP	876	–	–	–	–
ADP	881	0.00145	0.60%	0.45%	0.99%
Postpone	908	0.03205	3.76%	1.98%	9.41%
DirectShipment	1033	0.02040	12.23%	8.52%	18.66%

Table 4. Comparison between ADP and benchmark policies for larger instances

Primary vehicles	Max. arrivals per time unit	Max. inventory	Earliest dispatch	Costs ADP (normalized)	Costs DirectShipment	Costs Postpone
2	10	20	$\{0\}$	100	109.2	109.7
8	10	20	$\{0\}$	100	112.3	113.6
3	15	15	$\{0\}$	100	111.6	112.1
5	15	30	$\{0\}$	100	111.9	113.0
5	15	30	$\{0, 1\}$	100	113.5	113.9
5	15	30	$\{0, 1, 2\}$	100	114.9	115.8

to both DP and the benchmark policies. For all simulations, we use the same realizations of order arrivals, and perform 10,000 simulation replications over the planning horizon. We do this for ten initial states, selected to represent a variety of properties. Table 3 shows the comparison between DP, ADP, and the two benchmark policies. The percentages indicate the average difference in costs between the optimal solution and the simulation results. By applying our ADP policy, we incur costs that are on average 0.60% higher than the optimal solution, as such outperforming both benchmark policies. Also, the standard deviation in solution quality is much lower than for the benchmark policies. With ADP, we postpone 24% less orders than 'Postpone' does. For the initial states where 'DirectShipment' actually postpones orders – for some initial states it never will – ADP postpones 203% more.

Finally, we perform tests on six larger instances, with 10 customers, 10 order sizes, a maximum dispatch window length of 2, and a time horizon of 10. Tunable parameters are mentioned in Table 4. The size of the state space follows from the multinomial coefficients for all possible combinations of order arrivals, and is $\gg 10^{30}$ for all these instances. Clearly, an exact benchmark for such instances cannot be provided.

Table 4 shows the results of our experiments on the six larger instances. When the size of the action space exceeds a predefined threshold, we only partially enumerate the action space based on customer locations. We subsequently apply the same priority rules as the heuristic to assign orders to a given action. On average, ADP outperforms the policy 'DirectShipment' by 12.23% and the policy 'Postpone' by 13.02%. The results show how incorporating future information (both deterministic and stochastic) improves dispatching decisions. In the case of pre-announced orders, the outperformance is stronger due the myopic nature of the benchmark policies.

7 Conclusions

In this paper, we proposed an ADP approach to make dispatch decisions at urban consolidation centers. We optimized decisions for a finite planning horizon, taking into account stochastic order arrivals during this horizon. We have shown that ADP is able to closely approximate the optimal values obtained by DP for toy-sized instances of our problem. For larger instances, ADP clearly and consistently outperforms both myopic benchmark policies, indicating the added value of considering future information.

The ADP approach as described in this paper resolves the intractability of the state space and outcome space. However, we have not thoroughly addressed the size of the action space, which in the worst case increases exponentially with the number of orders in the system. A possible approach to tackle this problem – without affecting the quality of decision-making – is to express the single-period, single-state decision problem as an integer linear program, that can be solved to optimality with less computational effort. This requires the basis functions in the VFA to be defined in such a way that they are linear with the decision problem. Additionally, heuristic approaches to reduce the action space – as touched upon in this paper – are considered for future research.

Our numerical experiments have shown that even for small instances, simple consolidation policies that ignore future information are inadequate to capture the complexity of waiting decisions. Further research will focus on the evaluation of realistically-sized instances and comparison with more sophisticated benchmark policies. The basis functions as proposed in this paper may not work well on every instance. Insights in appropriate VFAs for a variety of problem structures will yield a valuable contribution to existing literature. Both the ADP approach and its benchmarks need to be refined in order to provide an in-depth analysis of the applicability of ADP on realistic-sized dispatch problems.

References

1. Bookbinder, J.H., Cai, Q., He, Q.M.: Shipment consolidation by private carrier: the discrete time and discrete quantity case. Stochastic Models **27**(4), 664–686 (2011)
2. Coelho, L.C., Laporte, G., Cordeau, J.F.: Dynamic and stochastic inventory-routing. Technical Report CIRRELT 2012–37, CIRRELT (2012)
3. Coelho, L.C., Cordeau, J.F., Laporte, G.: Thirty years of inventory routing. Transportation Science **48**(1), 1–19 (2014)
4. Daganzo, C.F.: The distance traveled to visit n points with a maximum of c stops per vehicle: An analytic model and an application. Transportation Science **18**(4), 331–350 (1984)
5. Lium, A.G., Crainic, T.G., Wallace, S.W.: A study of demand stochasticity in service network design. Transportation Science **43**(2), 144–157 (2009)

6. Minkoff, A.S.: A markov decision model and decomposition heuristic for dynamic vehicle dispatching. Operations Research **41**(1), 77–90 (1993)
7. Mutlu, F., Çetinkaya, S., Bookbinder, J.: An analytical model for computing the optimal time-and-quantity-based policy for consolidated shipments. IIE Transactions **42**(5), 367–377 (2010)
8. Pillac, V., Gendreau, M., Guéret, C., Medaglia, A.L.: A review of dynamic vehicle routing problems. European Journal of Operational Research **225**(1), 1–11 (2013)
9. Powell, W.B.: Approximate Dynamic Programming: Solving the Curses of Dimensionality, vol. 842. John Wiley & Sons (2011)
10. Powell, W.B., Topaloglu, H.: Stochastic programming in transportation and logistics. Handbooks in Operations Research and Management Science **10**, 555–635 (2003)
11. Quak, H.: Sustainability of urban freight transport: Retail distribution and local regulations in cities. No. EPS-2008-124-LIS. Erasmus Research Institute of Management (ERIM) (2008)
12. Ritzinger, U., Puchinger, J., Hartl, R.F.: A survey on dynamic and stochastic vehicle routing problems. International Journal of Production Research, 1–17 (2015). (ahead-of-print)
13. Robusté, F., Estrada, M., López-Pita, A.: Formulas for estimating average distance traveled in vehicle routing problems in elliptic zones. Transportation Research Record: Journal of the Transportation Research Board **1873**(1), 64–69 (2004)
14. SteadieSeifi, M., Dellaert, N., Nuijten, W., Van Woensel, T., Raoufi, R.: Multimodal freight transportation planning: A literature review. European Journal of Operational Research **233**(1), 1–15 (2014)
15. Topaloglu, H., Powell, W.B.: Dynamic-programming approximations for stochastic time-staged integer multicommodity-flow problems. INFORMS Journal on Computing **18**(1), 31–42 (2006)