A Parallelizable and Approximate Dynamic Programming-Based Dynamic Fleet Management Model with Random Travel Times and Multiple Vehicle Types

Huseyin Topaloglu

School of Operations Research and Industrial Engineering
Cornell University, Ithaca, NY 14853, USA
topaloglu@orie.cornell.edu
Tel:1-607-255-0698 Fax:1-607-255-9129

July 30, 2006

Abstract

This chapter presents an approximate dynamic programming-based dynamic fleet management model that can handle random load arrivals, random travel times and multiple vehicle types. Our model decomposes the fleet management problem into a sequence of time-indexed subproblems by formulating it as a dynamic program and uses approximations of the value function. To handle random travel times, the state variable of our dynamic program includes all individual decisions over a relevant portion of the history. We propose a sampling-based strategy to approximate the value function under this high-dimensional state variable in a tractable manner. Under our value function approximation strategy, the fleet management problem decomposes into a sequence of time-indexed min-cost network flow subproblems that naturally yield integer solutions. Moreover, the subproblem for each time period further decomposes by the locations, making our model suitable for parallel computing. Computational experiments show that our model yields high-quality solutions within reasonable runtimes.

1 Introduction and Relevant Literature

Although the majority of the dynamic fleet management models assume that the travel times are deterministic, there are a variety of applications where traffic jams, equipment failures and undesirable weather conditions create substantial variability in the travel times. Furthermore, even if these events are rare, the travel times may appear to be random to the modeler, since they depend on factors outside the scope of the model, such as the skill level of the drivers and the schedules of the ferryboats that are used by the vehicles to cross waterways. This chapter presents an approximate dynamic programming-based model for the dynamic fleet management problem with random load arrivals, random travel times and multiple vehicle types.

The work we present in this chapter is motivated by the empty railcar allocation setting. In the car allocation business, the railroad company receives car requests from its clients on a daily basis. These requests are for a particular number of cars of a particular type, at a particular operating station and on a particular date. The company decides which cars should be used to satisfy the requests and tries to get these cars to the clients. After using the cars for a certain amount of time, the clients return the cars to the company. To serve the clients in a prompt manner and to offset the imbalances between where the requests originate and where the cars are returned, the company continuously repositions the empty cars. Due to limited train capacities and shifting local train schedules, the travel times can be highly variable.

The strategy that we propose in this chapter has ties with the previous research. Godfrey and Powell (2002*a*) and Godfrey and Powell (2002*b*) propose approximate dynamic programming-based models for fleet management problems with random load arrivals, deterministic travel times and a single vehicle type. Topaloglu and Powell (2006) extend this work to problems with multiple vehicle types. The idea in these models is to decompose the fleet management problem into time-indexed subproblems by formulating it as a dynamic program and to use approximations of the value function. We employ a similar strategy here but we use a new dynamic programming formulation to handle random travel times and multiple vehicle types. The difficulty in handling random travel times arises from the fact that when a vehicle is dispatched from a particular origin to a particular destination, it is not known when the vehicle will reach its destination. Consequently, the state variable in our dynamic programming formulation keeps track of *all individual decisions over a relevant portion of the history*. This increases the number of dimensions of the state variable but we show that one can approximate the value function in a tractable manner under this high-dimensional state variable.

In two recent companion papers (see Topaloglu and Powell (2006) and Topaloglu (2005)), we address random load arrivals and random travel times in more restricted settings. One of our goals

here is to extend these papers and other earlier work in the following four dimensions to build fleet management models that can simultaneously handle random load arrivals, random travel times and multiple vehicle types. 1) We devise a value function approximation strategy under which the subproblems that need to be solved for each time period reduce to min-cost network flow problems that yield integer solutions naturally. If one naively attempts to generalize the earlier models to handle multiple vehicle types, then the subproblems that need to be solved for each time period reduce to min-cost integer multicommodity network flow problems, in which case obtaining integer solutions may be difficult. 2) We use separable approximations of the value function and there are other fleet management models that use such value function approximations. One can argue that separable approximations work well because the fleet management problem is "inherently separable" by the geographical locations due to the fact that the vehicles located at different locations can serve different sets of loads. However, it is difficult to claim that the fleet management problem is "inherently separable" by the vehicle types when there are multiple types of vehicles that can exist at the same location and compete to serve the same set of demands. In this case, the success of separable approximations is not as obvious. Our computational experiments indicate that separable approximations can work well even in the presence of multiple vehicle types. 3) Our model decomposes the fleet management problem by locations as well as by time periods. In particular, it solves one subproblem for each time period-location pair, and in a certain time period, the subproblems corresponding to different locations can be solved in parallel. When coupled with the fact that these subproblems are min-cost network flow problems, this parallelization opportunity gives our model a significant runtime advantage. Even for problems with deterministic load arrivals or deterministic travel times or a single vehicle type, our model may be preferable due to its runtime advantage. 4) As a byproduct of parallelization, making the decisions for different locations by solving independent subproblems accurately mimics the decision-making process in many applications. Freight carriers usually have multiple dispatchers responsible from managing the vehicles at different locations and each dispatcher pays little attention to the other dispatchers when making its vehicle allocation decisions. Our model allows each dispatcher to concentrate only on the location that it is responsible from and the dispatchers coordinate their decisions through the value function approximations.

Fleet management models have a long history and comprehensive reviews can be found in Dejax and Crainic (1987), Powell (1988), Powell, Jaillet and Odoni (1995) and Crainic and Laporte (1998). We restrict our review to the most relevant literature. Early fleet management models appear as the first applications of linear programming and min-cost network flow algorithms (see Dantzig and Fulkerson (1954), Ferguson and Dantzig (1955), White and Bomberault (1969) and White (1972)). These models formulate the problem over a *state-time network*, where the nodes represent the supply of vehicles at different locations and at different time periods, and the arcs represent the vehicle

movements. They assume that the load arrivals over the entire planning horizon are known in advance or incorporate the uncertain future load arrivals through their expected values. In practice, solving these models often requires integer programming techniques because the network structure is quickly lost when one attempts to address multiple vehicle types or load pick up windows (see Abara (1989) and Hane, Barnhart, Johnson, Marsten, Nemhauser and Sigismondi (1995)). Not too far from the state-time network models are the myopic assignment models that solve a simple assignment problem for each time period (see Powell (1988) and Powell (1996)). These models do not incorporate the uncertain future load arrivals at all and only work with what is known with certainty, which is justified by the fact that the carriers receive the shipment requests far in advance. They are easy to implement, and especially for this reason, they are widely used in practice.

A second class of fleet management models attempt to address the randomness in the load arrivals explicitly. The earliest examples of these models assume that a constant fraction, say β_{ijt}, of the empty vehicles available at location i at time period t is repositioned to location j. In this case, the fleet management problem can be formulated as a nonlinear program to find the best values for these fractions (see Jordan and Turnquist (1983) and Powell (1986)). Recent models address the randomness in the load arrivals by decomposing the problem into time-indexed subproblems and assessing the impact of the current decisions on the future through value functions. Due to the large number of decision variables and possible load realizations, classical dynamic programming techniques are not feasible for computing the value functions and most of the effort revolves around approximating the value functions in a tractable manner (see Frantzeskakis and Powell (1990), Crainic, Gendreau and Dejax (1993), Carvalho and Powell (2000), Godfrey and Powell (2002*a*), Godfrey and Powell (2002*b*) and Adelman (2004)). The model we present in this chapter falls in this category.

Myopic assignment models remain applicable when the travel times are random. Other than these straightforward models, we are not aware of a fleet management model that can handle random travel times. Laporte, Louveaux and Mercure (1992) and Kenyon and Morton (2003) consider random travel times in the context of the vehicle routing problem. Their work does not apply to the fleet management problem because they focus on building fixed vehicle routes that yield the best average performance, whereas the fleet management setting requires continuous management of the vehicles. Parallelization and distributed computing have not seen much attention in the area of transportation. These concepts usually appear as a byproduct of an algorithmic strategy such as Lagrangian relaxation or Dantzig-Wolfe decomposition. For example, Chien, Balakrishnan and Wong (1989) and Fumero and Vercellis (1999) propose decomposition strategies for inventory distribution problems motivated by Lagrangian relaxation. Bourbeau, Crainic and Gendron (2000) parallelize branch-and-bound for a large-scale fleet management application.

2 Problem Description

We have a heterogeneous fleet of vehicles to serve the loads that occur at different locations in a transportation network over a finite planning horizon. At every time period, a random number of loads enter the system, and we need to decide which loads we should carry and to which locations we should reposition the empty vehicles. We are interested in maximizing the total expected profit over the planning horizon. We define the following.

$\mathcal{T} =$ Set of time periods in the planning horizon, $\mathcal{T} = \{1, \ldots, T\}$ for some finite T.

$\mathcal{V} =$ Set of vehicle types.

$\mathcal{I} =$ Set of locations in the transportation network.

$\mathcal{L} =$ Set of movement modes using which a vehicle can move from one location to another, $\mathcal{L} = \{0, 1, \ldots, L\}$ for some finite L. Movement mode 0 always corresponds to empty repositioning. The other modes correspond to carrying different types of loads. We elaborate on the concept of movement modes below.

$x^v_{ijlt} =$ Number of vehicles of type v dispatched from location i to j at time period t under movement mode l.

$c^v_{ijlt} =$ Profit from dispatching one vehicle of type v from location i to j at time period t under movement mode l.

$D_{ijlt} =$ Random variable for the number of loads that need to be carried from location i to j at time period t and that correspond to movement mode l.

$\tau_{ij} =$ Random variable for the number of time periods required to move from location i to j. We assume that $1 \leq \tau_{ij} \leq \tau$ for some finite τ.

Whenever we have $l \in \{1, \ldots, L\}$, the decision variable x^v_{ijlt} captures the number of vehicles of type v carrying a load of type l from location i to j at time period t, and the profit from each one of these loads is c^v_{ijlt}. If it is not feasible to use a vehicle of type v to carry a load of type l, then we assume that $c^v_{ijlt} = -\infty$ for all $i, j \in \mathcal{I}$, $t \in \mathcal{T}$. The decision variable x^v_{ij0t} captures the number of vehicles of type v moving empty from location i to j at time period t, and the cost of each one of these movements is $-c^v_{ij0t}$. Since the empty movements are not bounded, we have $D_{ij0t} = \infty$ for all $i, j \in \mathcal{I}$, $t \in \mathcal{T}$. One advantage of our notation is that it does not require making a distinction between empty and loaded movements. For example, we can succinctly write the profit at time period t as $\sum_{i,j \in \mathcal{I}} \sum_{l \in \mathcal{L}} \sum_{v \in \mathcal{V}} c^v_{ijlt} \, x^v_{ijlt}$ and the number of vehicles of type v leaving location i at time period t as $\sum_{j \in \mathcal{I}} \sum_{l \in \mathcal{L}} x^v_{ijlt}$. Finally, we note that the decision variable x^v_{ii0t} captures the number of vehicles

of type v held at location i at time period t. For notational uniformity, we let $\tau_{ii} = 1$ for all $i \in \mathcal{I}$, although the travel time from a location to itself is, of course, 0 in reality.

By suppressing some of the indices in the variables above, we denote a vector composed of the components ranging over the suppressed indices. For example, we have $x_t = \{x^v_{ijlt} : i, j \in \mathcal{I},\ l \in \mathcal{L},\ v \in \mathcal{V}\}$ and $D_t = \{D_{ijlt} : i, j \in \mathcal{I},\ l \in \mathcal{L}\}$. We reserve the letters $s,\ t$ and u to index the time periods, and if two or three of them are used in the same context, then the ordering $s \leq t \leq u$ holds.

3 Model Formulation

We begin by reviewing the fleet management model proposed by Topaloglu and Powell (2006). Although it is unable to handle random travel times, this model gives a good starting point.

3.1 Deterministic Travel Times

To capture the state of the vehicles, we define the following.

$r^v_{iut} =$ Right before making the decisions at time period t, the number of vehicles of type v that are inbound to location i and that will reach location i at time period u.

Since $\tau_{ij} \leq \tau$ for all $i, j \in \mathcal{I}$, a vehicle dispatched to location i before time period t reaches its destination before time period $t+\tau$. Therefore, we have $r^v_{iut} = 0$ for all $i \in \mathcal{I},\ v \in \mathcal{V},\ u = t+\tau, \ldots, T$, and the vector $r_t = \{r^v_{iut} : i \in \mathcal{I},\ v \in \mathcal{V},\ u = t, \ldots, t+\tau-1\}$ completely defines the state of the vehicles right before making the decisions at time period t. We note that $r_1 = \{r^v_{iu1} : i \in \mathcal{I},\ v \in \mathcal{V},\ u = 1, \ldots, \tau\}$ gives the initial position of the vehicles and is a part of the problem data. Due to the decisions made before the beginning of the planning horizon of the problem, r^v_{iu1} can be greater than 0 for some $i \in \mathcal{I}$, $v \in \mathcal{V}$, $u > 1$.

Since r^v_{itt} captures the number of vehicles of type v available at location i at time period t, the set of feasible decisions for any state vector r_t and load realizations D_t is given by

$$\mathcal{X}(r_t, D_t) = \Big\{ x_t : \sum_{j \in \mathcal{I}} \sum_{l \in \mathcal{L}} x^v_{ijlt} = r^v_{itt} \quad i \in \mathcal{I},\ v \in \mathcal{V} \tag{1}$$

$$\sum_{v \in \mathcal{V}} x^v_{ijlt} \leq D_{ijlt} \qquad i, j \in \mathcal{I},\ l \in \mathcal{L} \tag{2}$$

$$x^v_{ijlt} \in \mathbb{Z}_+ \qquad i, j \in \mathcal{I},\ l \in \mathcal{L},\ v \in \mathcal{V} \Big\}, \tag{3}$$

where the left side of (1) accounts for the total number of vehicles of type v leaving location i and the left side of (2) accounts for the total number of vehicles carrying a load of type l from location i

to j. Given the decisions x_t and the state vector r_t at time period t, the state of the system at the beginning of the next time period is given by

$$r^v_{ju,t+1} = \sum_{i\in\mathcal{I}}\sum_{l\in\mathcal{L}} \mathbf{1}_{\tau_{ij}}(u-t)\; x^v_{ijlt} + r^v_{jut} \qquad j\in\mathcal{I},\; v\in\mathcal{V},\; u=t+1,\ldots,t+\tau, \tag{4}$$

where we assume that τ_{ij} is deterministic, and $\mathbf{1}_a(b)$ takes value 1 when $a=b$ and takes value 0 otherwise. We bring (1)-(4) together by

$$\mathcal{Y}(r_t,D_t) = \Big\{(x_t,r_{t+1}):\; x_t\in\mathcal{X}(r_t,D_t) \\ r^v_{ju,t+1} = \sum_{i\in\mathcal{I}}\sum_{l\in\mathcal{L}} \mathbf{1}_{\tau_{ij}}(u-t)\; x^v_{ijlt} + r^v_{jut} \quad j\in\mathcal{I},\; v\in\mathcal{V},\; u=t+1,\ldots,t+\tau\Big\},$$

whereby $(x_t,r_{t+1})\in\mathcal{Y}(r_t,D_t)$ means that the decisions x_t are feasible when the state of the system is r_t and the realization of the loads is D_t, and applying the decisions x_t on the state vector r_t generates the state vector r_{t+1} for the next time period. Using r_t as the state variable, the problem can be formulated as a dynamic program as

$$V_t(r_t) = \mathbb{E}\left\{\max_{(x_t,r_{t+1})\in\mathcal{Y}(r_t,D_t)} c_t\cdot x_t + V_{t+1}(r_{t+1}) \mid r_t\right\}, \tag{5}$$

where $V_t(\cdot)$ is the value function at time period t (see Puterman (1994)). For any system state r_t and load realizations D_t, the optimal decisions for time period t can be found by solving the *subproblem*

$$V_t(r_t,D_t) = \max_{(x_t,r_{t+1})\in\mathcal{Y}(r_t,D_t)} c_t\cdot x_t + V_{t+1}(r_{t+1}). \tag{6}$$

Due to the so-called curse of dimensionality, solving (5) in order to compute $\{V_t(\cdot): t\in\mathcal{T}\}$ is usually intractable. Motivated by the fact that the value function is piecewise-linear concave, Topaloglu and Powell (2006) propose replacing the value function $V_t(\cdot)$ with a separable piecewise-linear concave approximation $\hat{V}_t(\cdot)$ of the form

$$\hat{V}_t(r_t) = \sum_{i\in\mathcal{I}}\sum_{v\in\mathcal{V}}\sum_{u=t}^{t+\tau-1} \hat{V}^v_{iut}(r^v_{iut}), \tag{7}$$

where each $\hat{V}^v_{iut}(\cdot)$ is a single-dimensional piecewise-linear concave function. Consequently, they propose solving the *approximate subproblem*

$$\widetilde{V}_t(r_t,D_t) = \max_{(x_t,r_{t+1})\in\mathcal{Y}(r_t,D_t)} c_t\cdot x_t + \hat{V}_{t+1}(r_{t+1}) \tag{8}$$

to make the decisions at time period t, where $\widetilde{V}_t(r_t,D_t)$ is simply a place-holder for the optimal objective value. Their approach solves the problem above for different values of r_t and D_t, and iteratively improves the quality of the value function approximations. Our strategy closely parallels this approach, however we use a new state variable and a new method to improve the quality of the value function approximations. Closing this section, we note that, due to constraints (4), solving problem (8) requires prior knowledge of $\mathbf{1}_{\tau_{ij}}(u-t)$ for all $i,j\in\mathcal{I}$, $u=t+1,\ldots,t+\tau$. Therefore, this model cannot be used when the travel times are random.

3.2 Random Travel Times

We deal with the random travel times by keeping track of all individual decisions over a relevant portion of the history. To formalize, we define the following.

f^v_{ijst} = Right before observing the vehicle arrivals and making the decisions at time period t, the number of vehicles of type v that were dispatched from location i to j at time period s and that have not reached location j before time period t.

Since $\tau_{ij} \leq \tau$ for all $i, j \in \mathcal{I}$, a vehicle dispatched from location i to j before time period $t - \tau$ will reach its destination before time period t. Therefore, we have $f^v_{ijst} = 0$ for all $i, j \in \mathcal{I}$, $v \in \mathcal{V}$, $s = 1, \ldots, t-\tau-1$, and the vector $f_t = \{f^v_{ijst} : i, j \in \mathcal{I},\ v \in \mathcal{V},\ s = t-\tau, \ldots, t-1\}$ gives all decisions over the portion of the history relevant to the decisions made at time period t. Thus, we use f_t as the state variable in our dynamic programming formulation. We note that $f_1 = \{f^v_{ijs1} : i, j \in \mathcal{I},\ v \in \mathcal{V},\ s = 1-\tau, \ldots, 0\}$ gives the decisions made before the beginning of the planning horizon of the problem and is a part of the problem data.

Although the vector f_t captures the history relevant to the decisions made at time period t, the number of vehicles available at each location at time period t depends on the realizations of the travel times and is still a random variable. We define the following random variable.

A^v_{ijst} = Random variable representing the number of vehicles of type v that were dispatched from location i to j at time period s and that reach location j at time period t.

We note that f^v_{ijst} captures the number of vehicles of type v that were dispatched from location i to j at time period s and that have not reached location j before time period t, whereas A^v_{ijst} captures what "portion" of these vehicles actually reach location j at time period t. Therefore, we always have $A^v_{ijst} \leq f^v_{ijst}$. Section 5 gives a careful characterization of possible probability laws that may govern the random vector $A_t = \{A^v_{ijst} : i, j \in \mathcal{I},\ v \in \mathcal{V},\ s = t-\tau, \ldots, t-1\}$. For now, we view A_t as a random vector just like D_t whose value becomes known at the beginning of time period t.

Since $\sum_{j \in \mathcal{I}} \sum_{s=t-\tau}^{t-1} A^v_{jist}$ is the number of vehicles of type v available at location i at time period t, the set of feasible decisions for any state vector f_t, arrival realizations A_t and load realizations D_t is given by

$$\mathcal{X}(f_t, A_t, D_t) = \Big\{ x_t : \sum_{j \in \mathcal{I}} \sum_{l \in \mathcal{L}} x^v_{ijlt} = \sum_{j \in \mathcal{I}} \sum_{s=t-\tau}^{t-1} A^v_{jist} \quad i \in \mathcal{I},\ v \in \mathcal{V} \tag{9}$$
$$(2), (3) \Big\}.$$

Given the decisions x_t and the state vector f_t at time period t, the state of the system at the beginning of the next time period is given by

$$f^v_{ijt,t+1} = \sum_{l\in\mathcal{L}} x^v_{ijlt} \qquad i,j \in \mathcal{I},\ v \in \mathcal{V} \tag{10}$$

$$f^v_{ijs,t+1} = f^v_{ijst} - A^v_{ijst} \quad i,j \in \mathcal{I},\ v \in \mathcal{V},\ s = t+1-\tau,\dots,t-1. \tag{11}$$

In alignment with the definition of $f^v_{ijt,t+1}$, the right side of (10) computes the number of vehicles of type v dispatched from location i to j at time period t, whereas the right side of (11) computes what "portion" of the vehicles of type v that were dispatched from location i to j at time period s still remain in-transit after observing the arrivals at time period t. We bring (9)-(11) together by

$$\begin{aligned}\mathcal{Y}(f_t, A_t, D_t) = \Big\{(x_t, f_{t+1}) : x_t \in \mathcal{X}(f_t, A_t, D_t)\\ f^v_{ijt,t+1} = \sum_{l\in\mathcal{L}} x^v_{ijlt} \qquad i,j \in \mathcal{I},\ v \in \mathcal{V}\\ f^v_{ijs,t+1} = f^v_{ijst} - A^v_{ijst} \quad i,j \in \mathcal{I},\ v \in \mathcal{V},\ s = t+1-\tau,\dots,t-1\Big\}.\end{aligned}$$

Using f_t as the state variable, the problem can be formulated as a dynamic program as

$$V_t(f_t) = \mathbb{E}\left\{\max_{(x_t,f_{t+1})\in\mathcal{Y}(f_t,A_t,D_t)} c_t \cdot x_t + V_{t+1}(f_{t+1}) \mid f_t\right\}. \tag{12}$$

For any system state f_t, arrival realizations A_t and load realizations D_t, the optimal decisions for time period t can be found by solving the subproblem

$$V_t(f_t, A_t, D_t) = \max_{(x_t,f_{t+1})\in\mathcal{Y}(f_t,A_t,D_t)} c_t \cdot x_t + V_{t+1}(f_{t+1}). \tag{13}$$

We propose using separable value function approximations of the form

$$\hat{V}_t(f_t) = \sum_{i,j\in\mathcal{I}} \sum_{v\in\mathcal{V}} \sum_{s=t-\tau}^{t-1} \hat{V}^v_{ijst}(f^v_{ijst}), \tag{14}$$

where each *value function approximation component* $\hat{V}^v_{ijst}(\cdot)$ is a single-dimensional piecewise-linear concave function with points of nondifferentiability being a subset of integers. The approximation in (14) seems more complicated than the one in (7), but the next section shows that the approximate subproblem can be simplified to a great extent due to the separability of the approximation.

4 Structure of the Approximate Subproblems and Parallelization

Replacing the value function $V_{t+1}(\cdot)$ in (13) by an approximation of form (14), the approximate subproblem for time period t can be written as

$$\begin{aligned}\widetilde{V}_t(f_t, A_t, D_t) = \max_{x_t, f_{t+1}} \quad & \sum_{i,j\in\mathcal{I}} \sum_{l\in\mathcal{L}} \sum_{v\in\mathcal{V}} c^v_{ijlt}\, x^v_{ijlt} + \sum_{i,j\in\mathcal{I}} \sum_{v\in\mathcal{V}} \sum_{s=t+1-\tau}^{t} \hat{V}^v_{ijs,t+1}(f^v_{ijs,t+1}) \\ \text{subject to} \quad & (2), (3), (9), (10) \\ & f^v_{ijs,t+1} = f^v_{ijst} - A^v_{ijst} \qquad i,j \in \mathcal{I},\ v \in \mathcal{V},\ s = t+1-\tau,\dots,t-1.\end{aligned} \tag{15}$$

For the model with deterministic travel times in Section 3.1, Topaloglu and Powell (2006) show that the approximate subproblem (8) is a min-cost integer multicommodity network flow problem. Since the approximate subproblem (8) "spans" only one time period, it is a small min-cost integer multicommodity network flow problem, but the multicommodity characteristics still make it difficulty to obtain integer solutions and bring an unwelcome dimension of complexity. In this section, we show that the approximate subproblem (15) is a min-cost network flow problem.

We note that the last set of constraints in problem (15) set the decision variables $\{f^v_{ijs,t+1} : i,j \in \mathcal{I},\ v \in \mathcal{V},\ s = t+1-\tau,\ldots,t-1\}$ to constants. Therefore, by plugging their values in the objective function, we can drop these decision variables and the value function approximation components corresponding to them. This reduces problem (15) to

$$\widetilde{V}_t(f_t, A_t, D_t) = \max_{x_t, f_{t+1}} \sum_{i,j\in\mathcal{I}} \sum_{l\in\mathcal{L}} \sum_{v\in\mathcal{V}} c^v_{ijlt}\, x^v_{ijlt} + \sum_{i,j\in\mathcal{I}} \sum_{v\in\mathcal{V}} \hat{V}^v_{ijt,t+1}(f^v_{ijs,t+1}) \tag{16}$$
$$\text{subject to} \quad (2), (3), (9), (10).$$

Section 6 uses the next remark for updating and improving the value function approximations.

Remark 1. Noting constraints (10), the decision variable $f^v_{ijt,t+1}$ captures the number of vehicles of type v dispatched from location i to j at time period t. Therefore, $\hat{V}^v_{ijt,t+1}(f)$ can be interpreted as the approximation to the expected future benefit from dispatching f vehicles from location i to j at time period t.

Letting R be the total number of available vehicles, the relevant domain of $\hat{V}^v_{ijt,t+1}(\cdot)$ is $\{0,1,\ldots,R\}$ and we can represent $\hat{V}^v_{ijt,t+1}(\cdot)$ by a sequence of numbers $\{\eta^v_{ijt,t+1}(r) : r = 1,\ldots,R\}$, where $\eta^v_{ijt,t+1}(r)$ is the slope of $\hat{V}^v_{ijt,t+1}(\cdot)$ over $(r-1, r)$. That is, we have $\eta^v_{ijt,t+1}(r) = \hat{V}^v_{ijt,t+1}(r) - \hat{V}^v_{ijt,t+1}(r-1)$. In this case, we can write problem (16) explicitly as

$$\widetilde{V}_t(f_t, A_t, D_t) = \max_{x_t, f_{t+1}, z_{t+1}} \sum_{i,j\in\mathcal{I}} \sum_{l\in\mathcal{L}} \sum_{v\in\mathcal{V}} c^v_{ijlt}\, x^v_{ijlt} + \sum_{i,j\in\mathcal{I}} \sum_{v\in\mathcal{V}} \sum_{r=1}^{R} \eta^v_{ijt,t+1}(r)\ z^v_{ijt,t+1}(r) \tag{17}$$

$$\text{subject to} \sum_{j\in\mathcal{I}} \sum_{l\in\mathcal{L}} x^v_{ijlt} = \sum_{j\in\mathcal{I}} \sum_{s=t-\tau}^{t-1} A^v_{jist} \quad i\in\mathcal{I},\ v\in\mathcal{V} \tag{18}$$

$$\sum_{l\in\mathcal{L}} x^v_{ijlt} - f^v_{ijt,t+1} = 0 \qquad i,j\in\mathcal{I},\ v\in\mathcal{V} \tag{19}$$

$$f^v_{ijt,t+1} - \sum_{r=1}^{R} z^v_{ijt,t+1}(r) = 0 \qquad i,j\in\mathcal{I},\ v\in\mathcal{V} \tag{20}$$

$$\sum_{v\in\mathcal{V}} x^v_{ijlt} \le D_{ijlt} \qquad i,j\in\mathcal{I},\ l\in\mathcal{L} \tag{21}$$

$$z^v_{ijt,t+1}(r) \le 1 \qquad i,j\in\mathcal{I},\ v\in\mathcal{V},\ r = 1,\ldots,R \tag{22}$$

$$x^v_{ijlt},\ z^v_{ijt,t+1}(r) \in \mathbb{Z}_+ \qquad i,j\in\mathcal{I},\ l\in\mathcal{L},\ v\in\mathcal{V},\ r = 1,\ldots,R. \tag{23}$$

Defining three sets of nodes $\mathcal{N}_1 = \{(i,v) : i \in \mathcal{I},\ v \in \mathcal{V}\}$, $\mathcal{N}_2 = \{(i,j,v) : i,j \in \mathcal{I},\ v \in \mathcal{V}\}$ and $\mathcal{N}_3 = \{(i,j,v) : i,j \in \mathcal{I},\ v \in \mathcal{V}\}$, the problem above can be visualized as a min-cost integer multicommodity network flow problem that takes place over a network with the set of nodes $\mathcal{N}_1 \bigcup \mathcal{N}_2 \bigcup \mathcal{N}_3 \bigcup \{\Phi\}$, where Φ denotes a dummy root note. In this network, there exists an arc corresponding to each decision variable in problem (17), and Table 1 shows the "tail" and "head" nodes for each one of these arcs. Pictorially, problem (17) is the min-cost integer multicommodity network flow problem shown in Figure 1, where we assume that $\mathcal{I} = \{i_1, i_2\}$, $\mathcal{L} = \{l_1\}$, $\mathcal{V} = \{v_1, v_2\}$ and we label the nodes by $(i,v) \in \mathcal{I} \times \mathcal{V}$, $(i,j,v) \in \mathcal{I}^2 \times \mathcal{V}$. Constraints (18), (19) and (20) are respectively the flow balance constraints for the nodes in $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_3$. The flow balance constraint for node Φ is redundant and is not included in problem (17). Constraints (21) put a limit on the total flow over a set of arcs and seemingly give problem (17) multicommodity characteristics. However, as the next proposition shows, a simple transformation converts problem (17) into a min-cost network flow problem.

Proposition 1. Problem (17) can be solved as a min-cost network flow problem.

Proof. We combine constraints (19) and (20) into $\sum_{l\in\mathcal{L}} x^v_{ijlt} = \sum_{r=1}^{R} z^v_{ijt,t+1}(r)$ to drop the decision variables $f^v_{ijt,t+1}$ for all $i,j \in \mathcal{I}$, $v \in \mathcal{V}$. Adding the combined constraints for all $j \in \mathcal{I}$, we can write constraints (18) as $\sum_{j\in\mathcal{I}} \sum_{r=1}^{R} z^v_{ijt,t+1}(r) = \sum_{j\in\mathcal{I}} \sum_{s=t-\tau}^{t-1} A^v_{jist}$ for all $i \in \mathcal{I}$, $v \in \mathcal{V}$. We define new decision variables $\{y_{ijlt} : i,j \in \mathcal{I},\ l \in \mathcal{L}\}$ and split constraints (21) into $\sum_{v\in\mathcal{V}} x^v_{ijlt} - y_{ijlt} = 0$ and $y_{ijlt} \leq D_{ijlt}$ for all $i,j \in \mathcal{I}$, $l \in \mathcal{L}$. Therefore, problem (17) can be written as

$$\widetilde{V}_t(f_t, A_t, D_t) = \max_{x_t, z_{t+1}, y_t} \quad \sum_{i,j\in\mathcal{I}} \sum_{l\in\mathcal{L}} \sum_{v\in\mathcal{V}} c^v_{ijlt}\, x^v_{ijlt} + \sum_{i,j\in\mathcal{I}} \sum_{v\in\mathcal{V}} \sum_{r=1}^{R} \eta^v_{ijt,t+1}(r)\, z^v_{ijt,t+1}(r) \tag{24}$$

$$\text{subject to} \quad \sum_{j\in\mathcal{I}} \sum_{r=1}^{R} z^v_{ijt,t+1}(r) = \sum_{j\in\mathcal{I}} \sum_{s=t-\tau}^{t-1} A^v_{jist} \qquad i \in \mathcal{I},\ v \in \mathcal{V} \tag{25}$$

$$\sum_{l\in\mathcal{L}} x^v_{ijlt} - \sum_{r=1}^{R} z^v_{ijt,t+1}(r) = 0 \qquad i,j \in \mathcal{I},\ v \in \mathcal{V} \tag{26}$$

$$\sum_{v\in\mathcal{V}} x^v_{ijlt} - y_{ijlt} = 0 \qquad i,j \in \mathcal{I},\ l \in \mathcal{L} \tag{27}$$

$$y_{ijlt} \leq D_{ijlt} \qquad i,j \in \mathcal{I},\ l \in \mathcal{L}$$

$$(22), (23).$$

Defining three sets of nodes $\mathcal{O}_1 = \{(i,v) : i \in \mathcal{I},\ v \in \mathcal{V}\}$, $\mathcal{O}_2 = \{(i,j,v) : i,j \in \mathcal{I},\ v \in \mathcal{V}\}$ and $\mathcal{O}_3 = \{(i,j,l) : i,j \in \mathcal{I},\ l \in \mathcal{L}\}$, problem (24) is a min-cost network flow problem that takes place over a network with the set of nodes $\mathcal{O}_1 \bigcup \mathcal{O}_2 \bigcup \mathcal{O}_3 \bigcup \{\Phi\}$. Table 2 shows the "tail" and "head" nodes for each one of the arcs corresponding to the decision variables in problem (24). Constraints (25), (26) and (27) are respectively the flow balance constraints for the nodes in $\mathcal{O}_1$, $\mathcal{O}_2$ and $\mathcal{O}_3$. Figure 2 shows the general structure of problem (24), where we label the nodes by $(i,j) \in \mathcal{I} \times \mathcal{V}$,

$(i,j,v) \in \mathcal{I}^2 \times \mathcal{V}$, $(i,j,l) \in \mathcal{I}^2 \times \mathcal{L}$. □

The next remark will be useful for updating and improving the value function approximations.

Remark 2. Since problem (17) can be solved as a min-cost network flow problem, we can speak of the dual solution to problem (17).

The fact that problem (17) naturally yields integer solutions gives our model a dramatic runtime advantage. Furthermore, the objective function and the constraints of problem (17) decompose by $i \in \mathcal{I}$, which implies that one can obtain a solution to problem (17) by solving $|\mathcal{I}|$ smaller subproblems and these $|\mathcal{I}|$ subproblems can be solved in parallel. This parallelization opportunity further boosts the runtime advantage of our model.

5 Characterizing the Arrival Random Variables

In this section, we describe two possible models for the arrival random variables. The first model assumes that the travel times for different vehicles are independent, whereas the second model assumes that the travel times for different vehicles traveling between the same origin-destination pair are perfectly dependent.

Independent Model. This model assumes that the travel times for different vehicles are independent. Given that a particular vehicle was dispatched from location i to j at time period s and it has not reached location j before time period t, the probability that this vehicle reaches location j at time period t is $\mathbb{P}\{\tau_{ij} = t-s \,|\, \tau_{ij} \geq t-s\}$. The number of vehicles of type v that were dispatched from location i to j at time period s and that have not reached location j before time period t is given by f^v_{ijst}. A "portion" of these vehicles, which is captured by A^v_{ijst}, reach location j at time period t. Therefore, under the independent model, the number of vehicles of type v that were dispatched from location i to j at time period s and that reach location j at time period t is binomially distributed with parameters f^v_{ijst} and $\mathbb{P}\{\tau_{ij} = t-s \mid \tau_{ij} \geq t-s\}$. In other words, we have

$$\mathbb{P}\{A^v_{ijst} = a \,|\, f^v_{ijst} = f\} = \left(\begin{array}{c} f \\ a \end{array} \right) \Big[\mathbb{P}\{\tau_{ij} = t-s \,|\, \tau_{ij} \geq t-s\}\Big]^a \Big[\mathbb{P}\{\tau_{ij} > t-s \,|\, \tau_{ij} \geq t-s\}\Big]^{f-a}$$

for all $i,j \in \mathcal{I}$, $v \in \mathcal{V}$, $s = t-\tau,\ldots,t-1$ and the random variables $\{A^v_{ijst} : i,j \in \mathcal{I},\ v \in \mathcal{V},\ s = t-\tau,\ldots,t-1\}$ are independent. This model is applicable when the travel times depend on conditions internal to the vehicles or the drivers, such as breakdowns and skill levels.

Perfectly Dependent Model. This model assumes that all vehicles dispatched from location i to j at time period s reach location j at the same time period. There are f^v_{ijst} vehicles of type v that were dispatched from location i to j at time period s and that have not reached location j before

time period t. Under this model, either all of these vehicles reach location j at time period t (this happens with probability $\mathbb{P}\{\tau_{ij} = t - s \,|\, \tau_{ij} \geq t - s\}$) or none of these vehicles reach location j at time period t (this happens with probability $\mathbb{P}\{\tau_{ij} > t - s \,|\, \tau_{ij} \geq t - s\}$). Therefore, we have

$$\mathbb{P}\{A_{ijst} = \alpha \,|\, f_{ijst} = \phi\} = \begin{cases} 1 & \text{if } \alpha = 0,\ \phi = 0 \\ \mathbb{P}\{\tau_{ij} = t - s \,|\, \tau_{ij} \geq t - s\} & \text{if } \alpha = \phi,\ \phi \neq 0 \\ \mathbb{P}\{\tau_{ij} > t - s \,|\, \tau_{ij} \geq t - s\} & \text{if } \alpha = 0,\ \phi \neq 0 \\ 0 & \text{otherwise,} \end{cases}$$

where $A_{ijst} = \{A^v_{ijst} : v \in \mathcal{V}\}$, $f_{ijst} = \{f^v_{ijst} : v \in \mathcal{V}\}$, α and ϕ are $|\mathcal{V}|$-dimensional vectors. We still assume that the random vectors $\{A_{ijst} : i, j \in \mathcal{I},\ s = t - \tau, \ldots, t - 1\}$ are independent. This model is applicable when the travel times depend on external conditions, such as weather and traffic.

The next remark will be useful for updating and improving the value function approximations.

Remark 3. Under both models, a vehicle dispatched from location i to j at time period s reaches its destination at time period t with probability $\mathbb{P}\{\tau_{ij} = t - s\}$. This property holds for the first model because its probability law is equivalent to using the distribution of τ_{ij} to sample an independent travel time for each vehicle dispatched from location i to j. The second model also satisfies this property because its probability law is equivalent to using the distribution of τ_{ij} to sample one travel time for all vehicles dispatched from location i to j at a particular time period.

Clearly, both characterizations of the arrival random variables presented in this section have limitations and the travel times may have more complex dependencies in reality. For example, we assume that the travel times for the vehicles that travel between different origin-destination pairs or that start their trip at different time periods are independent. However, one would expect the travel times for the vehicles leaving or arriving at the same or nearby locations at the same or nearby time periods to be strongly correlated, especially if the travel times are affected by external conditions, such as weather or traffic. Nevertheless, the two characterizations we give serve as a good starting point. Furthermore, Remark 3 is the only property of these characterizations that we use in the rest of the paper. Therefore, our model should work with other possible characterizations of the arrival random variables that satisfy Remark 3.

6 Updating and Improving the Value Function Approximations

The performance of the decisions made by solving approximate subproblems of form (15) depends on how "well" the value function approximations approximate the exact value function. We propose a sampling-based strategy that constructs the value function approximations in an iterative manner. In particular, we let $\{\hat{V}^n_t(\cdot) : t \in \mathcal{T}\}$ be the set of value function approximations at iteration n.

For each time period t in the planning horizon, we sample a realization of A_t and D_t, which we respectively denote by $\tilde{A}_t^n$ and $\tilde{D}_t^n$, and solve the approximate subproblem

$$(x_t^n, f_{t+1}^n) = \underset{(x_t, f_{t+1}) \in \mathcal{Y}(f_t^n, \tilde{A}_t^n, \tilde{D}_t^n)}{\operatorname{argmax}} \sum_{i,j \in \mathcal{I}} \sum_{l \in \mathcal{L}} \sum_{v \in \mathcal{V}} c_{ijlt}^v \, x_{ijlt}^v + \sum_{i,j \in \mathcal{I}} \sum_{v \in \mathcal{V}} \sum_{s=t+1-\tau}^{t} \hat{V}_{ijs,t+1}^{vn}(f_{ijs,t+1}^v), \tag{28}$$

where f_1^n is initialized to reflect the initial state of the system. We note that solving approximate subproblems of form (28) for all $t \in \mathcal{T}$ is equivalent to simulating the behavior of the policy characterized by the value function approximations $\{\hat{V}_t^n(\cdot) : t \in \mathcal{T}\}$ under arrival realizations $\{\tilde{A}_t^n : t \in \mathcal{T}\}$ and load realizations $\{\tilde{D}_t^n : t \in \mathcal{T}\}$. The idea is to use the primal-dual solutions to the approximate subproblems to update and improve the value function approximations. The heuristic method that we use for this purpose is based on the following observations.

Remark 4. Comparing the approximate subproblems in (15) and (16) shows that we only need the value function approximation components $\{\hat{V}_{ijt,t+1}^v(\cdot) : i, j \in \mathcal{I},\ v \in \mathcal{V},\ t \in \mathcal{T}\}$. The value function approximation components $\{\hat{V}_{ijs,t+1}^v(\cdot) : i, j \in \mathcal{I},\ v \in \mathcal{V},\ s = t+1-\tau, \ldots, t-1,\ t \in \mathcal{T}\}$ do not affect our decisions at all and we do not need to improve them.

Remark 5. At iteration n, we dispatch $f_{ijt,t+1}^{vn}$ vehicles of type v from location i to j at time period t (see (10) and (28)). Recalling the interpretation of $\hat{V}_{ijt,t+1}^v(\cdot)$ in Remark 1, this implies that the quantity $\eta_{ijt,t+1}^{vn}(f_{ijt,t+1}^{vn}+1) = \hat{V}_{ijt,t+1}^{vn}(f_{ijt,t+1}^{vn}+1) - \hat{V}_{ijt,t+1}^{vn}(f_{ijt,t+1}^{vn})$ approximates the expected future benefit from dispatching an additional vehicle of type v from location i to j at time period t.

Remark 6. Noting problem (16), the approximate subproblem solved at time period t is

$$\begin{aligned} \widetilde{V}_t^n(f_t^n, \tilde{A}_t^n, \tilde{D}_t^n) = \max_{x_t, f_{t+1}} \quad & \sum_{i,j \in \mathcal{I}} \sum_{l \in \mathcal{L}} \sum_{v \in \mathcal{V}} c_{ijlt}^v \, x_{ijlt}^v + \sum_{i,j \in \mathcal{I}} \sum_{v \in \mathcal{V}} \hat{V}_{ijt,t+1}^{vn}(f_{ijt,t+1}) \\ \text{subject to} \quad & \sum_{j \in \mathcal{I}} \sum_{l \in \mathcal{L}} x_{ijlt}^v = \sum_{j \in \mathcal{I}} \sum_{s=t-\tau}^{t-1} \tilde{A}_{jist}^{vn} \qquad i \in \mathcal{I},\ v \in \mathcal{V} \\ & (2), (3), (10). \end{aligned} \tag{29}$$

The expression on the right side of constraints (29) is the number of vehicles of different types available at different locations at time period t. Since we can speak of the dual solution to the problem above due to Remark 2, we let $\{\theta_{it}^{vn} : i \in \mathcal{I},\ v \in \mathcal{V}\}$ be the optimal values of the dual variables associated with these constraints at iteration n. Therefore, θ_{it}^{vn} gives an estimate of the expected benefit from having an additional vehicle of type v at location i at time period t.

Remark 7. Noting Remark 3, a vehicle dispatched from location i to j at time period t reaches location j at time period u with probability $\mathbb{P}\{\tau_{ij} = u - t\}$. In view of Remark 6, this implies that, at iteration n, we can use $\vartheta_{ijt}^{vn} = \sum_{u=t+1}^{t+\tau} \mathbb{P}\{\tau_{ij} = u - t\} \, \theta_{ju}^{vn}$ to estimate the expected future benefit from dispatching an additional vehicle of type v from location i to j at time period t.

We now put Remarks 4-7 together. At iteration n, we dispatch $f_{ijt,t+1}^{vn}$ vehicles of type v from location i to j at time period t and $\eta_{ijt,t+1}^{vn}(f_{ijt,t+1}^{vn}+1)$ approximates the expected future benefit from an additional vehicle of type v dispatched from location i to j at time period t (Remark 5). Through the solution of the approximate subproblems at iteration n, we estimate the same quantity by ϑ_{ijt}^{vn} (Remarks 6 and 7). We use this new information to update and improve the value function approximation component $\hat{V}_{ijt,t+1}^{vn}(\cdot)$ through the smoothing method in Figure 3. Step 1 in this figure smoothes the slope of $\hat{V}_{ijt,t+1}^{vn}(\cdot)$ at the relevant point by using the new information. After smoothing, the function $Q_{ijt,t+1}^{vn}(\cdot)$ characterized by the sequence of slopes $\{q_{ijt,t+1}^{vn}(r) : r = 1, \ldots, R\}$ is not necessarily concave. Therefore, Step 2 projects the function $Q_{ijt,t+1}^{vn}(\cdot)$ onto the set of single-dimensional piecewise-linear concave functions with points of nondifferentiability being a subset of integers. Constraints (34) ensure that the value function approximation component $\hat{V}_{ijt,t+1}^{v,n+1}(\cdot)$ at the next iteration is concave. This updating method is due to Powell, Ruszczynski and Topaloglu (2004). Figure 4 describes our complete solution methodology.

7 Computational Experiments

In this section, we test the quality of the solutions obtained by our model and investigate how the runtimes scale with different problem parameters.

7.1 Experimental Setup

We present results on three problem classes. The first problem class includes problems with deterministic load arrivals and deterministic travel times. These problems can be formulated as min-cost integer multicommodity network flow problems. The second problem class includes problems with random load arrivals and deterministic travel times. The model in Section 3.1 can be used as a benchmark for these problems. The third problem class includes problems with random load arrivals and random travel times. We use the so-called rolling horizon strategy and an extension of the model in Section 3.1 as benchmarks for these problems. We include problems with deterministic travel times in our experimental setup because there exist a variety of solution methods for these problems. This enables us to carefully test the performance of our model. All of the algorithms were coded in JAVA 1.4.1 and ran on a Pentium IV Windows XP PC with 2.4 GHz CPU and 1 GB RAM.

In our experimental setup, we generate one basic problem and modify its certain attributes to generate different test problems. All of our test problems involve 5 vehicle types and 6 movement modes, one of which corresponds to empty repositioning and the other 5 correspond to serving different types of loads. In practice, the number of load types can be on the order of hundreds, but

our model scales well with the number of load types since the number of dimensions of the state variable does not depend on the number of load types. We let $c^v_{ij0t} = -c_0\,\delta(i,j)$ for all $i,j \in \mathcal{I}$, $v \in \mathcal{V}$, $t \in \mathcal{T}$, where c_0 is the "per-mile" empty repositioning cost and $\delta(i,j)$ is the distance between locations i and j. In practice, empty repositioning costs are often incurred on a "per-mile" basis. Letting C be a 5×5-dimensional matrix, we let $c^v_{ijlt} = r\,\delta(i,j)\,C^v_l - c_0\,\delta(i,j)$ for all $i,j \in \mathcal{I}$, $l \in \{1,\ldots,5\}$, $v \in \mathcal{V}$, $t \in \mathcal{T}$, where r is the revenue applied on a "per-mile" basis and $C^v_l \in [0,1]$ is the (v,l)-th entry of the matrix C. Consequently, there are more and less suitable vehicle types for each load type. Table 3 gives the different values we use for the matrix C. In practice, C^v_l may capture the willingness of the dispatcher to use a vehicle of type v to cover a load of type l, rather than a monetary discounting factor. The number of loads for each origin-destination pair, movement mode and time period is sampled from the Poisson distribution with the appropriate mean. The Poisson assumption is reasonable in many practical settings. We use the method described in Godfrey and Powell (2002*a*) to generate test problems where the number of loads inbound to a particular location is negatively correlated with the number of loads outbound from that location. We expect these problems to require plenty of empty repositioning movements in the optimal solution and naive solution methods should not give satisfactory results for them. We use $(T, |\mathcal{I}|, f, D, c_0, C) \in \{30,60,90\}\times\{10,20,40\}\times\{100,200,400\}\times\{2000,4000,6000\}\times\{1.6,4,8\}\times\{C_1,C_2,C_3,C_4\}$ to denote the attributes of our test problems, where the six dimensions respectively describe the number of time periods in the planning horizon, the number of locations in the transportation network, the size of the fleet, the expected number of loads over the planning horizon, the "per-mile" empty repositioning cost and the matrix characterizing the compatibility between the vehicle and load types. Our basic test problem, which is the first test problem reported in Tables 4, 5 and 6, involves 60 time periods, 20 locations, 200 vehicles, 4000 loads, empty repositioning cost of 4 and matrix C_1 from Table 3.

7.2 Computational Results

This section describes our computational results on the three aforementioned problem classes.

Problems with Deterministic Load Arrivals and Deterministic Travel Times. These problems can be formulated as a min-cost integer multicommodity network flow problem as

$$\begin{aligned}
\max_{x} \quad & \sum_{t\in\mathcal{T}}\sum_{i,j\in\mathcal{I}}\sum_{l\in\mathcal{L}}\sum_{v\in\mathcal{V}} c^v_{ijlt}\,x^v_{ijlt} && && (30)\\
\text{subject to} \quad & -\sum_{\substack{j\in\mathcal{I}:\\ t-\tau_{ji}\geq 1}}\sum_{l\in\mathcal{L}} x^v_{jil,t-\tau_{ji}} + \sum_{j\in\mathcal{I}}\sum_{l\in\mathcal{L}} x^v_{ijlt} = \sum_{\substack{j\in\mathcal{I}:\\ t-\tau_{ji}\leq 0}} f^v_{ji,t-\tau_{ji},1} && i\in\mathcal{I},\ v\in\mathcal{V},\ t\in\mathcal{T}\\
& \sum_{v\in\mathcal{V}} x^v_{ijlt} \leq D_{ijlt} && i,j\in\mathcal{I}\ , l\in\mathcal{L},\ t\in\mathcal{T},
\end{aligned}$$

where we assume that the load arrivals and the travel times are deterministic, and we omit the

integrality and nonnegativity constraints for brevity. The first set of constraints in problem (30) are the flow balance constraints. The expression on their right side is the number of vehicles of type v that reach location i at time period t and that were dispatched before the beginning of the planning horizon. As mentioned in Section 3.2, $f_1 = \{f^v_{ijs1} : i, j \in \mathcal{I},\ v \in \mathcal{V},\ s = 1-\tau, \ldots, 0\}$ is a part of the problem data. We apply the algorithm in Figure 4 for 250 iterations and compare the objective value at the final iteration with the objective value of problem (30). Table 4 shows the ratio of the two objective values (multiplied by 100), along with the runtime per iteration of the algorithm in Figure 4. For majority of the test problems, our model yields results within 2% of the optimal solution. The runtime per iteration for our model increases linearly with the number of time periods and almost quadratically with the number of locations. We emphasize that the reported runtimes are for a serial implementation where the subproblems corresponding to different locations are solved sequentially. If one were to solve the subproblems corresponding to different locations in parallel, then the runtime per iteration would increase linearly with the number of locations as well. Figure 5 shows the progress of the objective value as a function of the iteration number for two test problems. The objective value increases smoothly over the first 50 iterations and stabilizes.

Problems with Random Load Arrivals and Deterministic Travel Times. For these problems, we use the model described in Section 3.1 as a benchmark. Topaloglu and Powell (2006) compare this model with a variety of benchmarks and report that it yields high-quality solutions.

In a stochastic setting, testing our model or the model described in Section 3.1 requires two sets of iterations. The first set of iterations, which we refer to as the training iterations, follow the algorithm in Figure 4 by solving the approximate subproblem (15) (or the approximate subproblem (8) if we are using the model in Section 3.1) for all time periods and updating the value function approximations. In the second set of iterations, which we refer to as the testing iterations, we stop updating the value function approximations and simply simulate the behavior of the policy characterized by the value function approximations obtained during the training iterations. The goal of the testing iterations is to evaluate the quality of the value function approximations that are obtained during the training iterations. We use 250 training iterations and 100 testing iterations.

In Table 5, the first column shows the ratio of the average objective values obtained in the testing iterations by our model and by the model described in Section 3.1 (multiplied by 100), whereas the second and third columns show the runtimes per iteration for the two models. For this problem class, our model lags behind the benchmark strategy with comparable runtimes. However, we emphasize that a parallel implementation speeds up the runtimes of our model by a factor of $|\mathcal{I}|$, but this is not possible for the model described in Section 3.1.

Problems with Random Load Arrivals and Random Travel Times. The first benchmark method we use for these problems is a common engineering practice called the rolling horizon strategy (see Topaloglu (2005)). This strategy assumes that future random variables will take on their expected values. It makes the decisions at time period t by solving a deterministic problem that "spans" the time periods $t, t+1, \ldots, t+N$, where N is the rolling horizon length. In particular, for a given state vector f_t, arrival realizations A_t and load realizations D_t at time period t, the N-period rolling horizon strategy makes the decisions at time period t by solving the problem

$$\max_{x} \quad \sum_{u=t}^{t+N} \sum_{i,j\in\mathcal{I}} \sum_{l\in\mathcal{L}} \sum_{v\in\mathcal{V}} c^v_{ijlu}\, x^v_{ijlu} \tag{31}$$

$$\text{subject to} \quad \sum_{j\in\mathcal{I}} \sum_{l\in\mathcal{L}} x^v_{ijlt} = \sum_{j\in\mathcal{I}} \sum_{s=t-\tau}^{t-1} A^v_{jist} \qquad i\in\mathcal{I},\ v\in\mathcal{V} \tag{32}$$

$$- \sum_{\substack{j\in\mathcal{I}:\\ u-\bar{\tau}_{ji}\geq t}} \sum_{l\in\mathcal{L}} x^v_{jil,u-\bar{\tau}_{ji}} + \sum_{j\in\mathcal{I}} \sum_{l\in\mathcal{L}} x^v_{ijlu} = \sum_{\substack{j\in\mathcal{I}:\\ u-\bar{\tau}_{ji}\leq t-1}} \left[f^v_{ji,u-\bar{\tau}_{ji},t} - A^v_{ji,u-\bar{\tau}_{ji},t} \right]$$

$$i\in\mathcal{I},\ v\in\mathcal{V},\ u=t+1,\ldots,t+N \tag{33}$$

$$\sum_{v\in\mathcal{V}} x^v_{ijlt} \leq D_{ijlt} \qquad i,j\in\mathcal{I},\ l\in\mathcal{L}$$

$$\sum_{v\in\mathcal{V}} x^v_{ijlu} \leq \mathbb{E}\{D_{ijlu}\} \qquad i,j\in\mathcal{I},\ l\in\mathcal{L},\ u=t+1,\ldots,t+N,$$

where we use $\bar{\tau}_{ij}$ to denote the expected value of τ_{ij}. The problem above includes decision variables for time periods $t, t+1, \ldots, t+N$, but we only implement the decisions corresponding to time period t and re-solve a similar problem when making the decisions for the next time period. Since $\sum_{j\in\mathcal{I}} \sum_{s=t-\tau}^{t-1} A^v_{jist}$ gives the number of vehicles of type v available at location i at time period t, constraints (32) state that the total number of vehicles of type v that leave location i at time period t equals the number of vehicles of type v that are available at location i at time period t. Constraints (33) are the flow balance constraints analogous to the first set of constraints in problem (30). The expression $f^v_{ji,u-\bar{\tau}_{ji},t} - A^v_{ji,u-\bar{\tau}_{ji},t}$ on their right side is the number of vehicles of type v that were dispatched from location j to i at time period $u-\bar{\tau}_{ji}$ and that have not reached location i after having observed the arrivals at time period t. The rolling horizon strategy assumes that these vehicles will reach location i at time period u $(= u - \bar{\tau}_{ji} + \bar{\tau}_{ji})$. Since we solve problem (31) after having observed the realizations of the random variables at time period t, the right side of constraints (33) involves known constants. The second benchmark strategy we use is a heuristic extension of the model described in Section 3.1. In particular, since this model cannot accommodate random travel times, we assume that the travel times take on their expected values and make the decisions accordingly. Topaloglu (2005) explains in detail how one can heuristically use the model described in Section 3.1 in the presence of random travel times. In Table 6, the first column shows the ratio of the average objective values obtained in the testing iterations by our model and by the rolling horizon

strategy, whereas the second column shows the ratio of the average objective values obtained in the testing iterations by our model and the model described in Section 3.1 (all ratios are multiplied by 100). The third, fourth and fifth columns show the runtimes per iteration for the three models. The results indicate that our model performs noticeably better than both strategies. Interestingly, the runtimes for the rolling strategy can depend on the choice of the matrix C, but this does not seem to be an issue for our model. Also, with increasing number of locations, it is clear that implementing the rolling horizon strategy will be difficult.

8 Conclusions and Research Prospects

This chapter presented a dynamic fleet management model that can handle random load arrivals, random travel times and multiple vehicle types. Computational experiments showed that our model provides high-quality solutions within reasonable runtimes. An important feature of our model is that it decomposes the fleet management problem by time periods and by locations. When there are multiple dispatchers responsible from managing the vehicles at different locations, this feature allows them to concentrate on the vehicle supplies only at their own locations and they can coordinate their decisions with the help of the value function approximations.

The fleet management context provides a rich and challenging research area. There has been much advance in the last two decades but many practical issues remain unresolved. For example, we are not aware of a large-scale fleet management model that can handle load pick up and delivery windows and advance load information in a satisfactory manner, especially when the load arrivals are random. We heuristically used our model with some success in the presence of load pick up windows by simply keeping the loads in the system as long as their pick up windows are not expired. Nevertheless, the sound way to address load pick up and delivery windows is to include the loads in the state variable in the dynamic programming formulation, but it is not clear how to approximate the value function when the state variable includes this extra load dimension. Other important issues that need attention are incorporating the terminal capacities, balancing the allocation of the vehicles over the network and the simultaneous management of different types of resources such as trucks, containers and drivers. The approximate dynamic programming paradigm combines the intelligence of optimization with the flexibility of simulation, and may provide remedies for some of these unresolved issues.

References

Abara, J. (1989), 'Applying integer linear programming to the fleet assignment problem', *Interfaces* **19**(4), 20–28.

Adelman, D. (2004), Price-directed control of a closed logistics queueing network, Technical report, The University of Chicago, Graduate School of Business.

Bourbeau, B., Crainic, T. G. and Gendron, B. (2000), 'Branch-and-bound parallelization strategies applied to depot location and container fleet management problem', *Parallel Computing* **26**(1), 27–46.

Carvalho, T. A. and Powell, W. B. (2000), 'A multiplier adjustment method for dynamic resource allocation problems', *Transportation Science* **34**, 150–164.

Chien, T. W., Balakrishnan, A. and Wong, R. T. (1989), 'An integrated inventory allocation and vehicle routing problem', *Transportation Science* **23**(2), 67–76.

Crainic, T. G., Gendreau, M. and Dejax, P. (1993), 'Dynamic and stochastic models for the allocation of empty containers', *Operations Research* **41**, 102–126.

Crainic, T. G. and Laporte, G., eds (1998), *Fleet Management and Logistics*, Kluver Academic Publishers.

Dantzig, G. and Fulkerson, D. (1954), 'Minimizing the number of tankers to meet a fixed schedule', *Naval Research Logistics Quarterly* **1**, 217–222.

Dejax, P. and Crainic, T. (1987), 'A review of empty flows and fleet management models in freight transportation', *Transportation Science* **21**, 227–247.

Ferguson, A. and Dantzig, G. B. (1955), 'The problem of routing aircraft - A mathematical solution', *Aeronautical Engineering Review* **14**, 51–55.

Frantzeskakis, L. and Powell, W. B. (1990), 'A successive linear approximation procedure for stochastic dynamic vehicle allocation problems', *Transportation Science* **24**(1), 40–57.

Fumero, F. and Vercellis, C. (1999), 'Synchronized development of production, inventory and distribution schedules', *Transportation Science* **33**(3), 330–340.

Godfrey, G. A. and Powell, W. B. (2002*a*), 'An adaptive, dynamic programming algorithm for stochastic resource allocation problems I: Single period travel times', *Transportation Science* **36**(1), 21–39.

Godfrey, G. A. and Powell, W. B. (2002*b*), 'An adaptive, dynamic programming algorithm for stochastic resource allocation problems II: Multi-period travel times', *Transportation Science* **36**(1), 40–54.

Hane, C. A., Barnhart, C., Johnson, E. L., Marsten, R. E., Nemhauser, G. L. and Sigismondi, G. (1995), 'The fleet assignment problem: Solving a large scale integer program', *Mathematical Programming* **70**, 211–232.

Jordan, W. and Turnquist, M. (1983), 'A stochastic dynamic network model for railroad car distribution', *Transportation Science* **17**, 123–145.

Kenyon, A. S. and Morton, D. P. (2003), 'Stochastic vehicle routing with random travel times', *Transportation Science* **37**(1), 69–82.

Laporte, G., Louveaux, F. and Mercure, H. (1992), 'The vehicle routing problem with stochastic travel times', *Transportation Science* **26**(3), 161–170.

Powell, W. B. (1986), 'A stochastic model of the dynamic vehicle allocation problem', *Transportation Science* **20**, 117–129.

Powell, W. B. (1988), A comparative review of alternative algorithms for the dynamic vehicle allocation problem, *in* B. Golden and A. Assad, eds, 'Vehicle Routing: Methods and Studies', North Holland, Amsterdam, pp. 249–292.

Powell, W. B. (1996), 'A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers', *Transportation Science* **30**(3), 195–219.

Powell, W. B., Jaillet, P. and Odoni, A. (1995), Stochastic and dynamic networks and routing, *in* C. Monma, T. Magnanti and M. Ball, eds, '*Handbook in Operations Research and Management Science*, Volume on *Networks*', North Holland, Amsterdam, pp. 141–295.

Powell, W. B., Ruszczynski, A. and Topaloglu, H. (2004), 'Learning algorithms for separable approximations of stochastic optimization problems', *Mathematics of Operations Research* **29**(4), 814–836.

Puterman, M. L. (1994), *Markov Decision Processes*, John Wiley and Sons, Inc., New York.

Topaloglu, H. (2005), A parallelizable dynamic fleet management model with random travel times, Technical report, Cornell University, School of Operations Research and Industrial Engineering.

Topaloglu, H. and Powell, W. B. (2006), 'Dynamic programming approximations for stochastic, time-staged integer multicommodity flow problems', *INFORMS Journal on Computing* **18**(1), 31–42.

White, W. (1972), 'Dynamic transshipment networks: An algorithm and its application to the distribution of empty containers', *Networks* **2**(3), 211–236.

White, W. and Bomberault, A. (1969), 'A network algorithm for empty freight car allocation', *IBM Systems Journal* **8**(2), 147–171.

9 List of Tables

Arc	"Tail" node	"Head" node
x^v_{ijlt}	$(i,v) \in \mathcal{N}_1$	$(i,j,v) \in \mathcal{N}_2$
$f^v_{ijt,t+1}$	$(i,j,v) \in \mathcal{N}_2$	$(i,j,v) \in \mathcal{N}_3$
$z^v_{ijt,t+1}(r)$	$(i,j,v) \in \mathcal{N}_3$	Φ

Table 1: Incidence relationships of the arcs corresponding to the decision variables in problem (17).

Arc	"Tail" node	"Head" node
$z^v_{ijt,t+1}(r)$	$(i,v) \in \mathcal{O}_1$	$(i,j,v) \in \mathcal{O}_2$
x^v_{ijlt}	$(i,j,v) \in \mathcal{O}_2$	$(i,j,l) \in \mathcal{O}_3$
y_{ijlt}	$(i,j,l) \in \mathcal{O}_3$	Φ

Table 2: Incidence relationships of the arcs corresponding to the decision variables in problem (24).

C_1				
1	0.8	0.5	0.3	0
0.7	1	0.8	0.3	0
0.6	0.6	1	0.5	0.5
0	0.4	0.7	1	0.5
0	0.4	0.6	0.6	1

C_2				
1	0	0	0	0
1	1	0	0	0
1	1	1	0	0
1	1	1	1	0
1	1	1	1	1

C_3				
1	0.5	0	0	0
0.5	1	0.5	0	0
0	0.5	1	0.5	0
0	0	0.5	1	0.5
0	0	0	0.5	1

C_4				
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Table 3: Matrices characterizing the compatibility between vehicle and load types.

Problem	Perf. rat.	CPU (sec.) our model
$(60, 20, 200, 4000, 4, C_1)$	99.04	3.7
$(30, 20, 200, 2000, 4, C_1)$	98.63	1.9
$(90, 20, 200, 6000, 4, C_1)$	98.76	6.0
$(60, 10, 200, 4000, 4, C_1)$	99.53	1.2
$(60, 40, 200, 4000, 4, C_1)$	98.97	14.3
$(60, 20, 200, 4000, 4, C_2)$	98.94	3.8
$(60, 20, 200, 4000, 4, C_3)$	98.83	3.6
$(60, 20, 200, 4000, 4, C_4)$	99.24	3.8
$(60, 20, 100, 4000, 4, C_1)$	97.09	3.1
$(60, 20, 400, 4000, 4, C_1)$	98.66	3.9
$(60, 20, 200, 4000, 1.6, C_1)$	98.71	3.8
$(60, 20, 200, 4000, 8, C_1)$	99.12	3.8

Table 4: Results for problems with deterministic load arrivals and deterministic travel times.

Problem	Perf. rat.	CPU (sec.) our model	CPU (sec.) model in §3.1
$(60, 20, 200, 4000, 4, C_1)$	96.09	3.7	3.4
$(30, 20, 200, 2000, 4, C_1)$	95.13	1.9	1.8
$(90, 20, 200, 6000, 4, C_1)$	92.40	6.0	5.7
$(60, 10, 200, 4000, 4, C_1)$	97.01	1.2	0.9
$(60, 40, 200, 4000, 4, C_1)$	95.62	14.3	10.1
$(60, 20, 200, 4000, 4, C_2)$	96.88	3.8	4.6
$(60, 20, 200, 4000, 4, C_3)$	95.16	3.6	2.9
$(60, 20, 200, 4000, 4, C_4)$	98.47	3.8	5.2
$(60, 20, 100, 4000, 4, C_1)$	95.78	3.1	3.3
$(60, 20, 400, 4000, 4, C_1)$	96.53	3.9	3.1
$(60, 20, 200, 4000, 1.6, C_1)$	95.39	3.8	3.7
$(60, 20, 200, 4000, 8, C_1)$	94.72	3.8	3.5

Table 5: Results for problems with random load arrivals and deterministic travel times.

Problem	Perf. rat. roll. hor.	Perf. rat. model in §3.1	CPU(sec.) our model	CPU (sec.) roll. hor.	CPU(sec.) model in §3.1
$(60, 20, 200, 4000, 4, C_1)$	105.21	103.84	3.7	237.1	3.4
$(30, 20, 200, 2000, 4, C_1)$	107.02	101.54	1.9	128.7	1.8
$(90, 20, 200, 6000, 4, C_1)$	103.17	103.28	6.0	334.9	5.7
$(60, 10, 200, 4000, 4, C_1)$	100.27	99.34	1.2	82.2	0.9
$(60, 40, 200, 4000, 4, C_1)$	105.92	104.29	14.3	1106.4	10.1
$(60, 20, 200, 4000, 4, C_2)$	103.85	102.17	3.8	1549.0	4.6
$(60, 20, 200, 4000, 4, C_3)$	108.05	101.14	3.6	233.5	2.9
$(60, 20, 200, 4000, 4, C_4)$	104.16	102.72	3.8	967.5	5.2
$(60, 20, 100, 4000, 4, C_1)$	106.89	105.99	3.1	274.9	3.3
$(60, 20, 400, 4000, 4, C_1)$	100.96	100.45	3.9	231.8	3.1
$(60, 20, 200, 4000, 1.6, C_1)$	104.15	103.61	3.8	228.3	3.7
$(60, 20, 200, 4000, 8, C_1)$	101.53	102.43	3.8	244.2	3.5

Table 6: Results for problems with random load arrivals and random travel times.

10 List of Figures

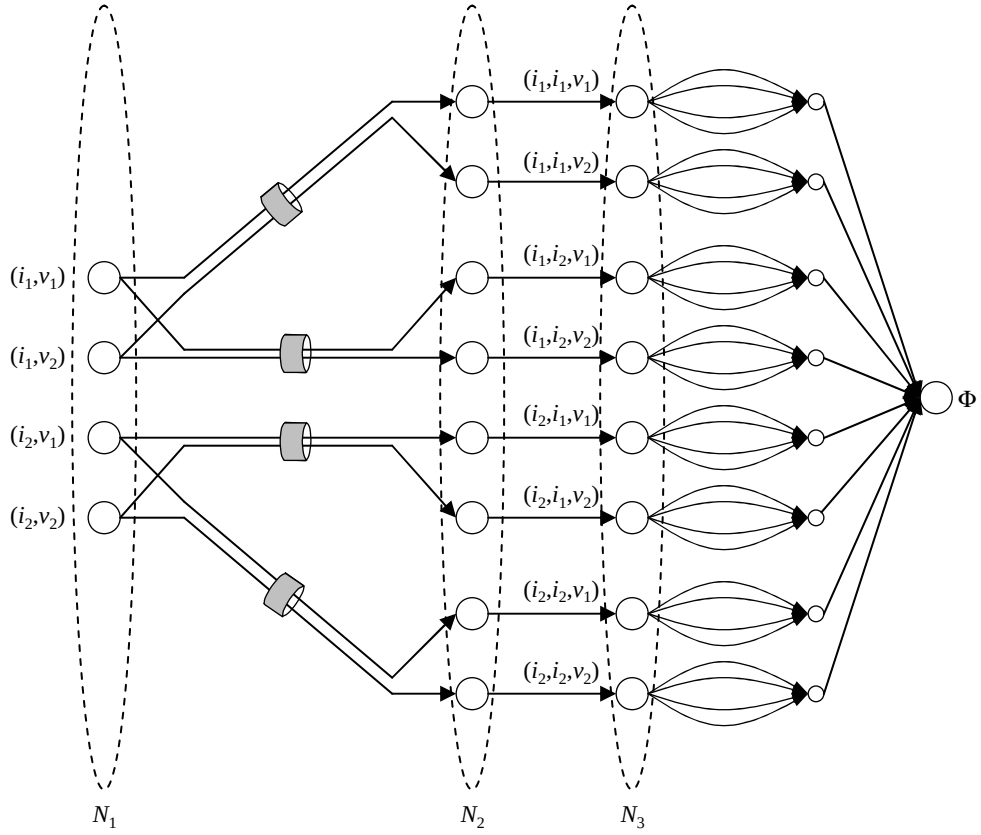

Figure 1: Problem (17) as a min-cost integer multicommodity network flow problem.

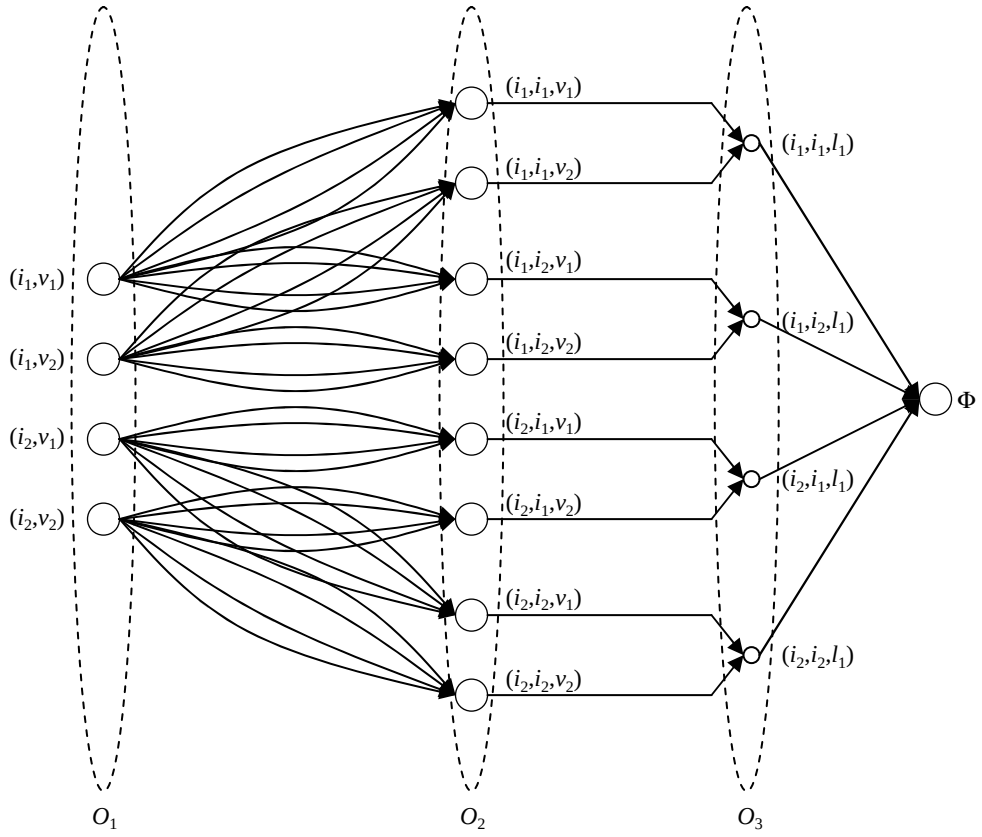

Figure 2: Problem (24) as a min-cost network flow problem.

Step 1. For all $r = 1, \ldots, R$, let

$$q^{vn}_{ijt,t+1}(r) = \begin{cases} (1-\alpha^n)\,\eta^{vn}_{ijt,t+1}(r) + \alpha^n\,\vartheta^{vn}_{ijt} & \text{if } r = f^n_{ijt,t+1} + 1 \\ \eta^{vn}_{ijt,t+1}(r) & \text{if } r \in \{1, \ldots, f^{vn}_{ijt,t+1}, f^{vn}_{ijt,t+1} + 2, \ldots, R\}, \end{cases}$$

where $\alpha^n \in [0,1]$ is the smoothing constant at iteration n.

Step 2. Let the vector $\eta^{v,n+1}_{ijt,t+1} = \{\eta^{v,n+1}_{ijt,t+1}(r) : r = 1, \ldots, R\}$, which characterizes the value function approximation component $\hat{V}^{v,n+1}_{ijt,t+1}(\cdot)$ at the next iteration $n+1$, be

$$\begin{aligned} \eta^{v,n+1}_{ijt,t+1} = \underset{z}{\text{argmin}} \quad & \sum_{r=1}^{R} \left[z(r) - q^{vn}_{ijt,t+1}(r)\right]^2 \\ \text{subject to} \quad & z(r) - z(r-1) \leq 0 \qquad r = 2, \ldots, R. \end{aligned} \tag{34}$$

Figure 3: A smoothing method to update the value function approximation component $\hat{V}^{vn}_{ijt,t+1}(\cdot)$.

Step 1. Initialize $n = 1$. Initialize $\{\hat{V}^{vn}_{ijt,t+1}(\cdot) : i, j \in \mathcal{I},\ v \in \mathcal{V},\ t \in \mathcal{T}\}$ to piecewise-linear concave functions with points of nondifferentiability being a subset of integers (possibly 0).

Step 2. Initialize $t = 1$. Initialize f^n_1 to reflect the initial state of the system.

Step 3. Given f^n_t, let $\tilde{A}^n_t$ and $\tilde{D}^n_t$ respectively be samples of A_t and D_t.

Step 4. Solve the approximate subproblem (15). Let

$$
\begin{aligned}
(x^n_t, f^n_{t+1}) = \underset{x_t, f_{t+1}}{\text{argmax}} \quad & \sum_{i,j\in\mathcal{I}} \sum_{l\in\mathcal{L}} \sum_{v\in\mathcal{V}} c^v_{ijlt}\, x^v_{ijlt} + \sum_{i,j\in\mathcal{I}} \sum_{v\in\mathcal{V}} \sum_{s=t+1-\tau}^{t} \hat{V}^{vn}_{ijs,t+1}(f^v_{ijs,t+1}) \\
\text{subject to} \quad & \sum_{j\in\mathcal{I}} \sum_{l\in\mathcal{L}} x^v_{ijlt} = \sum_{j\in\mathcal{I}} \sum_{s=t-\tau}^{t-1} \tilde{A}^{vn}_{jist} \quad i \in \mathcal{I},\ v \in \mathcal{V} \qquad (35) \\
& \sum_{v\in\mathcal{V}} x^v_{ijlt} \le \tilde{D}^n_{ijlt} \qquad\qquad i, j \in \mathcal{I},\ l \in \mathcal{L} \\
& f^v_{ijs,t+1} = f^{vn}_{ijst} - \tilde{A}^{vn}_{ijst} \qquad\qquad i, j \in \mathcal{I},\ v \in \mathcal{V},\ s = t+1-\tau, \ldots, t-1 \\
& (3),\ (10).
\end{aligned}
$$

Let $\{\theta^{vn}_{it} : i \in \mathcal{I},\ v \in \mathcal{V}\}$ be the optimal values of the dual variables associated with constraints (35).

Step 5. Increase t by 1. If $t \le T$, then go to Step 3.

Step 6. For all $i, j \in \mathcal{I}$, $v \in \mathcal{V}$, $t \in \mathcal{T}$, let $\vartheta^{vn}_{ijt} = \sum_{u=t+1}^{t+\tau} \mathbb{P}\{\tau_{ij} = u - t\}\ \theta^{vn}_{ju}$.

Step 7. For all $i, j \in \mathcal{I}$, $v \in \mathcal{V}$, $t \in \mathcal{T}$, use ϑ^{vn}_{ijt} and $\eta^{vn}_{ijt,t+1} = \{\eta^{vn}_{ijt,t+1}(r) : r = 1, \ldots, R\}$ in the smoothing method in Figure 3 to obtain the value function approximation component $\hat{V}^{v,n+1}_{ijt,t+1}(\cdot)$ that will be used at the next iteration.

Step 8. Increase n by 1. If one more iteration is needed, then go to Step 2.

Figure 4: The complete solution method.

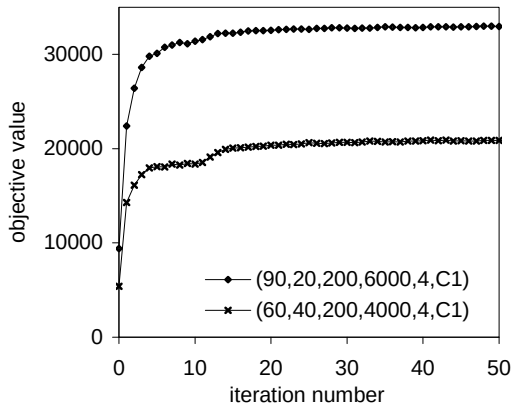

Figure 5: Progress of our model as a function of the iteration number for two test problems.