A Parallelizable Dynamic Fleet Management Model with Random Travel Times

Huseyin Topaloglu[*]

June 24, 2005

Abstract

In this paper, we present a stochastic model for the dynamic fleet management problem with random travel times. Our approach decomposes the problem into time-staged subproblems by formulating it as a dynamic program and uses approximations of the value function. In order to deal with random travel times, the state variable of our dynamic program includes *all individual decisions* over a relevant portion of the history. We show how to approximate the value function in a tractable manner under this new high-dimensional state variable.

Under our approximation scheme, the subproblem for each time period decomposes with respect to locations, making our model very appealing for large-scale applications. Numerical work shows that the proposed approach provides high-quality solutions and performs significantly better than standard benchmark methods.

Keywords: Transportation, Logistics, Approximate Dynamic Programming, Fleet Management, Distributed Decision-Making.

In dynamic fleet management models, travel time data can be highly variable due to breakdowns, weather conditions and traffic jams. Even if these events are rare, travel times may "appear" to be random to the modeler, since they depend on factors outside the scope of the model, such as time schedules of ferryboats and trains, nature of the freight, skill level of the driver. In this paper, we present a dynamic programming-based model for the dynamic fleet management problem (DFMP) with random travel times and random load arrivals.

Our work is motivated by an empty railcar allocation problem. In this problem, shippers place requests for cars at different locations on a daily basis. The railroad company decides which cars to use to satisfy a particular request and tries to get these cars to the shipper. After using them to transport goods between two or more locations, the shipper returns the cars to the railroad company, possibly at a different location. Depending on train capacities and local train schedules, travel times can be highly variable. Furthermore, the amount of time during which a shipper uses a car is almost always random. Therefore, a model that can deal with random travel times is crucial to capture the essence of the empty railcar allocation operations.

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

Although our work originated from a railcar allocation problem, the presentation in this paper is in the context of the DFMP. This considerably simplifies the notation and we believe that there is a larger audience familiar with the latter problem. Our approach is inspired by Godfrey and Powell (2002*a*) and Topaloglu and Powell (to appear), where the authors decompose the DFMP into time-staged subproblems by formulating it as a dynamic program, replace the value functions with separable approximations, and iteratively update and improve these approximations. We employ a similar approach here, however we use an entirely new state variable, dynamic programming formulation and improvement procedure in order to handle random travel times. The challenge in dealing with random travel times is that once we dispatch a vehicle from location a to b, we do not know when it will reach location b. For this reason, representing the state of the system at a certain time period requires keeping track of *all individual decisions* over a relevant portion of the history. Although this considerably increases the dimensionality of the state vector, we develop an approximation scheme under which the subproblem solved for each time period is tractable.

Besides handling random travel times, another advantage of our model is that it decomposes the DFMP by locations as well as by time periods. In particular, our model 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. There are several benefits from this decomposition. 1) Clearly, it gives our model a dramatic runtime advantage and makes our model suitable for large-scale applications. 2) Under our value function approximation scheme, the subproblem for each time period-location pair can be solved by a simple sort operation. 3) Most importantly, making the decisions for different locations by solving independent subproblems accurately reflects the real-world decision-making structure in many fleet management applications. Freight carriers have multiple dispatchers responsible for making the vehicle allocation decisions at different locations. When making their decisions, these dispatchers pay very little attention to the decisions of other dispatchers or the vehicle supplies in other locations. Our model allows each dispatcher to manage the vehicles in its own location by solving a "local" subproblem whose scope does not include the decisions of other dispatchers. The dispatchers at different locations coordinate their decisions through the value function approximations.

In this paper, we make the following research contributions. 1) We present a new dynamic programming formulation of the DFMP that can handle random travel times. In order to do so, we use a state variable that includes all individual decisions over a relevant portion of the history. We show how to approximate the value function of this dynamic program in a tractable manner. 2) We show that under our value function approximation scheme, the DFMP decomposes not only by time periods but also by locations, and the subproblems for different locations can be solved in parallel. Furthermore, solving these subproblems involves only a simple sort operation. 3) Numerical experiments show that our model performs significantly better than standard benchmark methods for problems with random travel times and provides near-optimal solutions for problems with deterministic data.

The rest of the paper is organized as follows. In Section 1, we briefly review the related literature. In Section 2, we describe the problem and introduce the necessary notation. Section 3 presents a new dynamic programming formulation of the DFMP and explains how to approximate the value function using separable, piecewise-linear, concave functions. Section 4 describes how we use samples of the load realizations and travel times to update and improve the value function approximations. In Section 5, we show that under our approximation scheme, the subproblem that needs to be solved for each time period decomposes by locations and the subproblem for each time period-location pair can be solved by a sort operation. Finally in Section 6, we present numerical results showing that our approach yields high-quality solutions and performs significantly better than standard benchmark methods.

1 Literature Review

In this section, we briefly review the related literature. Detailed surveys can be found in Dejax and Crainic (1987), Powell (1988), Powell, Jaillet and Odoni (1995).

We classify the fleet management models into three groups according to the way they handle the dynamic and stochastic nature of the problem. One class of models formulate the DFMP on a "state-time network," where the nodes represent the supply of vehicles in different states at different time periods and the arcs represent the vehicle movements (see, for example, Dantzig and Fulkerson (1954), Hane, Barnhart, Johnson, Marsten, Nemhauser and Sigismondi (1995), Holmberg, Joborn and Lundgren (1998)). The state of a vehicle may be as simple as its location or a combination of its location, size, maintenance level, etc. These models effectively deal with the dynamic nature of the problem and are simple to implement. However, their size grows very fast with the increasing number of locations and they easily lose their network structure if additional constraints are introduced. They incorporate the uncertain, future loads only through their expected values.

Myopic models that solve a simple assignment problem for each time period form a second class (see, for example, Powell (1988), Powell (1996)). These models only use the data that is known with certainty and make no effort to incorporate the uncertain, future loads. This is usually justified by the fact that the shippers tend to call in early and a large portion of the future loads are known in advance. Due to their simplicity and ease of implementation, they form the backbones of many commercial fleet management systems. An important shortcoming of these models is that they are unable to recommend where to reposition the vehicles in anticipation of future loads.

Finally, we have the truly stochastic models, which decompose the DFMP with respect to time periods and assess the impact of the current decisions on the future through a *recourse* or *value function* (see, for example, Frantzeskakis and Powell (1990), Crainic, Gendreau and Dejax (1993), Carvalho and Powell (2000), Godfrey and Powell (2002*a*), Godfrey and Powell (2002*b*), Topaloglu and Powell (to appear)). Due to high number of possible load realizations, large problem sizes and integrality requirements, general stochastic optimization approaches, such as scenario-based methods (see Wets (1974)) or backward dynamic programming (see Puterman (1994)), are not feasible for computing the value functions arising from practical DFMP instances. Therefore, most of the stochastic fleet management models revolve around the idea of approximating the value function in a tractable manner. The model that we present in this paper falls into this category. We formulate the DFMP as a dynamic program and use separable, piecewise-linear, concave approximations of the value function. The original features of our model are it can handle random travel times and it decomposes the DFMP not only with respect to time periods but also with respect to locations.

Other than myopic assignment models, which remain applicable when the travel times are random, we are not aware of a fleet management model that addresses the issue of 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 fails to apply to the DFMP setting since they focus on building apriori vehicle routes that yield the best average performance, whereas the DFMP calls for continuous management of the vehicles.

Decomposition, distributed problem solving and parallelization have been widely applied in areas such as manufacturing systems and allocation of computational resources (see, for example, Kurose and Simha (1989), Waldspurger, Hogg, Huberman, Kephart and Stornetta (1992), Albayrak and Krallmann (1995)). In the area of logistics, decomposition often appears as a result of using Lagrangian relaxation techniques to dualize certain constraints (see, for example, Chien, Balakrishnan and Wong (1989), Fumero and Vercellis (1999)). Bourbeau, Crainic and Gendron (2000) investigate parallelizing branch-and-bound techniques for large-scale fleet management applications.

2 Problem Description

We have a fleet of vehicles that can be used to serve the loads that occur at different locations over time. At every time period, a random number of loads enter the system, and we need to decide which loads we should serve and to which locations we should reposition the vehicles. We are interested in maximizing the total expected profit over a finite planning horizon. We assume that the fleet is homogenous and the loads that are not served in one time period are lost. (We briefly explain how to relax these assumptions later.) We define the following.

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

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

$\mathcal{K}_{ij}$ = Set of movement modes from location i to j, $\mathcal{K}_{ij} = \{0, \ldots, K_{ij}\}$. The movement mode 0 always corresponds to empty repositioning. The other modes in $\mathcal{K}_{ij}$ correspond to carrying different types of loads from location i to j.

x_{ijkt} = Number of vehicles dispatched from location i to j at time period t using movement mode k.

c_{ijkt} = Profit from dispatching one vehicle from location i to j at time period t using movement mode k.

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

τ_{ij} = Random variable representing the number of time periods required to move from location i to j. We assume $\tau_{ii} = 1$ and $1 \leq \tau_{ij} \leq \tau$ for some finite τ. The assumption that $\tau_{ii} = 1$ is for mathematical convenience. In reality, the travel time to move from a location to itself is, of course, 0.

Many fleet management models use two different decision variables, say x_{ijt} and y_{ijt}, to represent the numbers of empty repositioning and loaded movements from location i to j at time period t. In our notation, the number of empty repositioning movements from location i to j at time period t is captured by x_{ij0t}, whereas the number of loaded movements from location i to j at time period t is captured by $\{x_{ijkt} : k \in \mathcal{K}_{ij} \setminus \{0\}\}$. In practice, the movement modes in $\mathcal{K}_{ij} \setminus \{0\}$ may correspond to different types of loads or different shippers. Since the movement mode 0 corresponds to empty repositioning, we have $D_{ij0t} = \infty$ and $c_{ij0t} \leq 0$ for all $i, j \in \mathcal{L}$, $t \in \mathcal{T}$. The decision variable x_{ii0t} captures the number of vehicles held empty at location i at time period t. For brevity, we omit the dependence of the number of movement nodes on the origin and destination, and set $\mathcal{K}_{ij} = \mathcal{K}$, $K_{ij} = K$, for all $i, j \in \mathcal{L}$. It is also easy to extend our model to handle different travel time distributions for different movement modes.

By suppressing some of the indices in the variables and the parameters above, we denote a vector composed of the components ranging over the suppressed indices. For example, $x_t = \{x_{ijkt} : i, j \in \mathcal{L},\ k \in \mathcal{K}\}$, $D_t = \{D_{ijkt} : i, j \in \mathcal{L},\ k \in \mathcal{K}\}$. We reserve the letters s, t and u to index 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 Problem Formulation

We begin by briefly reviewing the dynamic programming formulation in Topaloglu and Powell (to appear). Although it is unable to handle random travel times, this formulation serves as a starting point.

3.1 Deterministic Travel Time Model

In this section we assume that τ_{ij} is deterministic for all $i, j \in \mathcal{L}$. We define

r_{iut} = Right before making the decisions for time period t, number of vehicles that are inbound to location i and that will arrive at location i at time period u.

Since $\tau_{ij} \leq \tau$ for all $i, j \in \mathcal{L}$, a vehicle dispatched to location i before time period t reaches its destination before time period $t + \tau$. Therefore, $r_{iut} = 0$ for all $i \in \mathcal{L},\ u = t + \tau, \dots, T$, and the vector $r_t = \{r_{iut} : i \in \mathcal{L},\ u = t, \dots, t + \tau - 1\}$ completely defines the state of the vehicles in order to make the decisions at time period t. For $t = 1$, $r_1 = \{r_{iu1} : i \in \mathcal{L},\ u = 1, \dots, \tau\}$ 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_{iu1} can be greater than 0 for some $u > 1$.

Noting that r_{itt} gives the number of vehicles available at location i at time period t, the set of feasible decisions for any state vector r_t and load realization D_t is given by

$$\mathcal{X}(r_t, D_t) = \Big\{ x_t : \sum_{j \in \mathcal{L}} \sum_{k \in \mathcal{K}} x_{ijkt} = r_{itt} \qquad \text{for all } i \in \mathcal{L} \tag{1}$$

$$x_{ijkt} \leq D_{ijkt} \qquad \text{for all } i, j \in \mathcal{L},\ k \in \mathcal{K} \tag{2}$$

$$x_{ijkt} \in \mathbb{Z}_+ \qquad \text{for all } i, j \in \mathcal{L},\ k \in \mathcal{K} \Big\}. \tag{3}$$

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 defined by the system dynamics equations

$$r_{ju,t+1} = \sum_{i \in \mathcal{L}} \sum_{k \in \mathcal{K}} \mathbf{1}_{\tau_{ij}}(u - t)\ x_{ijkt} + r_{jut} \qquad \text{for all } j \in \mathcal{L},\ u = t + 1, \dots, t + \tau, \tag{4}$$

where we use

$$\mathbf{1}_a(b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise.} \end{cases}$$

We also define

$$\mathcal{Y}(r_t, D_t) = \Big\{ (x_t, r_{t+1}) \ :\ x_t \in \mathcal{X}(r_t, D_t)$$
$$r_{ju,t+1} = \sum_{i \in \mathcal{L}} \sum_{k \in \mathcal{K}} \mathbf{1}_{\tau_{ij}}(u - t)\ x_{ijkt} + r_{jut} \quad \text{for all } j \in \mathcal{L},\ u = t + 1, \dots, t + \tau \Big\},$$

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 and denoting the value function for time period t as $V_t(\cdot)$, the problem can be formulated as a dynamic program as

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

(see Bellman (1957)). For any system state r_t and realization of loads D_t, the decisions for time period t are made by solving the subproblem

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

However, for almost all problem instances of practical significance, solving (5) in order to compute $\{V_t(\cdot) : t \in \mathcal{T}\}$ is intractable due to the well-known *curse of dimensionality.* Topaloglu and Powell (to appear) propose replacing the value functions with tractable approximations $\{\widehat{V}_t(\cdot) : t \in \mathcal{T}\}$ and solve an approximate subproblem

$$\widetilde{V}_t(r_t, D_t) = \max_{(x_t, r_{t+1}) \in \mathcal{Y}(r_t, D_t)} c_t x_t + \widehat{V}_{t+1}(r_{t+1}) \tag{7}$$

in order to make the decisions for time period t, where $\widetilde{V}_t(r_t, D_t)$ simply denotes the optimal objective value of the problem above. They solve problem (7) for different values of r_t and D_t, and iteratively improve the quality of the approximations. Motivated by the fact that the exact value function is piecewise-linear and concave, they use separable, piecewise-linear, concave value function approximations of the form

$$\widehat{V}_t(r_t) = \sum_{i \in \mathcal{L}} \sum_{u=t}^{t+\tau-1} \widehat{V}_{iut}(r_{iut}), \tag{8}$$

where each $\widehat{V}_{iut}(\cdot)$ is a single-dimensional, piecewise-linear, concave function. We note that when solving problem (7), constraints (4) require apriori knowledge of $\mathbf{1}_{\tau_{ij}}(u-t)$ for all $i, j \in \mathcal{L}$, $u = t+1, \ldots, t+\tau$. Therefore this model is not applicable when the travel times are random.

3.2 Random Travel Time Model

As mentioned in the introduction, when the travel times are random, we need to keep track of all individual decisions over a relevant portion of the history. In order to formalize the idea, we define

f_{ijst} = Right before observing the vehicles that reach their destinations at time period t, number of vehicles 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 $i, j \in \mathcal{L}$, a vehicle dispatched from location i to j before time period $t - \tau$ will reach its destination before time period t. Therefore, $f_{ijst} = 0$ for all $i, j \in \mathcal{L}$, $s = 1, \ldots, t - \tau - 1$, and the vector $f_t = \{f_{ijst} : i, j \in \mathcal{L},\ s = t - \tau, \ldots, t - 1\}$ gives all the decisions over the portion of the history relevant to the decisions made at time period t. As such, we use f_t as the state variable at time period t in our dynamic programming formulation. For $t = 1$, $f_1 = \{f_{ijs1} : i, j \in \mathcal{L},\ s = 1 - \tau, \ldots, 0\}$ gives the decisions made before the beginning of the planning horizon of the problem and is a part of the problem data. (If all the vehicles are idle at the beginning of the problem, we can set f_{ii01} to the initial vehicle supply at location i. The assumption that $\tau_{ii} = 1$ ensures that these vehicles will be available at location i at time period 1.)

The vector f_t captures all the past decisions relevant to the decisions made at time period t. However, the number of vehicles available at each location at time period t is still a random variable, since this quantity depends on travel times. We define

$A_{ijst} =$ Random variable representing the number of vehicles that were dispatched from location i to j at time period s and that arrive at location j at time period t.

The vector $A_t = \{A_{ijst} : i, j \in \mathcal{L},\ s = t - \tau, \ldots, t - 1\}$ gives all the arrival random variables that are of interest for the decisions made at time period t. At the end of this section, we give a careful characterization of possible probability laws that may govern this random variable. For the moment, we assume that A_t is a random vector just like D_t whose value becomes known at the beginning of time period t. (However the distribution of A_t has to depend on f_t, but this may not be the case for D_t.) The fact that the value of A_t becomes known at the beginning of time period t ensures that the travel time of a vehicle is not known until its trip is actually completed.

Then, the set of feasible decisions for any state vector f_t, arrival realization A_t and load realization D_t is given by

$$\mathcal{X}(f_t, A_t, D_t) = \Big\{ x_t : \ \sum_{j \in \mathcal{L}} \sum_{k \in \mathcal{K}} x_{ijkt} = \sum_{j \in \mathcal{L}} \sum_{s = t - \tau}^{t-1} A_{jist} \qquad \text{for all } i \in \mathcal{L} \tag{9}$$

$$x_{ijkt} \leq D_{ijkt} \qquad \text{for all } i, j \in \mathcal{L},\ k \in \mathcal{K} \tag{10}$$

$$x_{ijkt} \in \mathbb{Z}_+ \qquad \text{for all } i, j \in \mathcal{L},\ k \in \mathcal{K} \Big\}, \tag{11}$$

whereas the system dynamics are given by

$$f_{ijt,t+1} = \sum_{k \in \mathcal{K}} x_{ijkt} \qquad \text{for all } i, j \in \mathcal{L} \tag{12}$$

$$f_{ijs,t+1} = f_{ijst} - A_{ijst} \qquad \text{for all } i, j \in \mathcal{L},\ s = t + 1 - \tau, \ldots, t - 1. \tag{13}$$

We note that $f_{ijt,t+1}$ is "measured" before observing the vehicle arrivals at time period $t + 1$. As such, $f_{ijt,t+1}$ is the total number of vehicles dispatched from location i to j at time period t (see the system dynamics equations in (12)). A portion of these vehicles may, of course, reach location j at time period $t+1$. We also define

$$\mathcal{Y}(f_t, A_t, D_t) = \Big\{ (x_t, f_{t+1}) \ : \ x_t \in \mathcal{X}(f_t, A_t, D_t) \qquad f_{ijt,t+1} = \sum_{k \in \mathcal{K}} x_{ijkt} \quad \text{for all } i, j \in \mathcal{L} \qquad f_{ijs,t+1} = f_{ijst} - A_{ijst} \quad \text{for all } i, j \in \mathcal{L}, \ s = t+1-\tau, \ldots, t-1 \Big\}.$$

Using f_t as the state variable and denoting the value function for time period t as $V_t(\cdot)$, the problem can be formulated as a dynamic program as

$$V_t(f_t) = \mathbb{E} \Big\{ \max_{(x_t, f_{t+1}) \in \mathcal{Y}(f_t, A_t, D_t)} c_t x_t + V_{t+1}(f_{t+1}) \mid f_t \Big\}. \tag{14}$$

For any system state f_t, realization of arrivals A_t and realization of loads D_t, the decisions for time period t are made by solving the subproblem

$$V_t(f_t, A_t, D_t) = \max_{(x_t, f_{t+1}) \in \mathcal{Y}(f_t, A_t, D_t)} c_t x_t + V_{t+1}(f_{t+1}). \tag{15}$$

Similar to Section 3.1, we use separable value function approximations of form

$$\widehat{V}_t(f_t) = \sum_{i,j \in \mathcal{L}} \sum_{s=t-\tau}^{t-1} \widehat{V}_{ijst}(f_{ijst}), \tag{16}$$

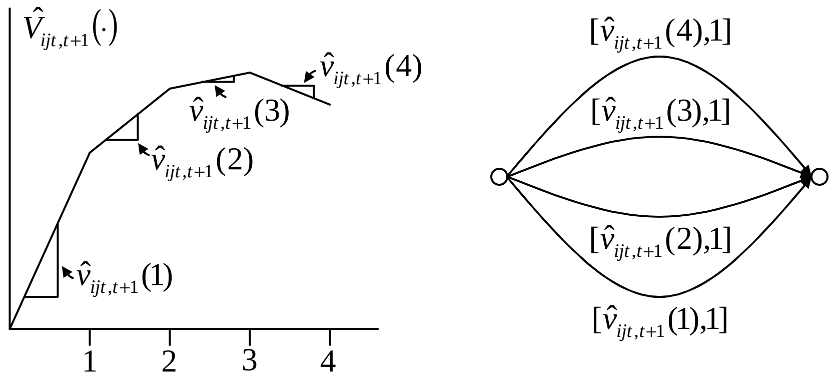
where each value function approximation component $\widehat{V}_{ijst}(\cdot)$ is a single-dimensional, piecewise-linear, concave function with points of nondifferentiability being a subset of integers. Approximation (16) seems more complicated than (8). However, we now show that the approximate subproblem can be simplified to a great extend due to the separability of the value function approximation. Under the new state variable, replacing the value function $V_{t+1}(\cdot)$ in (15) by an approximation of the form in (16), the approximate subproblem for time period t can explicitly be written as

$$\widetilde{V}_t(f_t, A_t, D_t) = \max_{x_t, f_{t+1}} \sum_{i,j \in \mathcal{L}} \sum_{k \in \mathcal{K}} c_{ijkt} \ x_{ijkt} + \sum_{i,j \in \mathcal{L}} \sum_{s=t+1-\tau}^{t} \widehat{V}_{ijs,t+1}(f_{ijs,t+1}) \tag{17}$$

$$\text{subject to} \quad (9), \ (10), \ (11), \ (12)$$
$$f_{ijs,t+1} = f_{ijst} - A_{ijst} \qquad \text{for all } i, j \in \mathcal{L}, \ s = t+1-\tau, \ldots, t-1.$$

In this problem, the last set of constraints simply set the variables $\{f_{ijs,t+1} : i, j \in \mathcal{L}, \ s = t+1-\tau, \ldots, t-1\}$ to constants. By plugging their values in the objective function, these variables and the value function approximation components corresponding to them can be dropped to get

$$\widetilde{V}_t(f_t, A_t, D_t) = \max_{x_t, f_{t+1}} \sum_{i,j \in \mathcal{L}} \sum_{k \in \mathcal{K}} c_{ijkt} \ x_{ijkt} + \sum_{i,j \in \mathcal{L}} \widehat{V}_{ijt,t+1}(f_{ijt,t+1}) \tag{18}$$
$$\text{subject to} \quad (9), \ (10), \ (11), \ (12).$$



$\hat{V}_{ijt,t+1}(\cdot)$
$\hat{v}_{ijt,t+1}(4)$
$\hat{v}_{ijt,t+1}(3)$
$\hat{v}_{ijt,t+1}(2)$
$\hat{v}_{ijt,t+1}(1)$
1
2
3
4
$[\hat{v}_{ijt,t+1}(4),1]$
$[\hat{v}_{ijt,t+1}(3),1]$
$[\hat{v}_{ijt,t+1}(2),1]$
$[\hat{v}_{ijt,t+1}(1),1]$

Figure 1: Piecewise-linear, concave objective functions can be incorporated into min-cost network flow problems by using a series of parallel arcs. (We use the notation [cost, upper bound].)

Assuming that R is the total number of available vehicles, the relevant domain of $\widehat{V}_{ijt,t+1}(\cdot)$ is $\{0, 1, \ldots, R\}$. In this case, we can represent $\widehat{V}_{ijt,t+1}(\cdot)$ by a sequence of numbers $\{\hat{v}_{ijt,t+1}(r) : r = 1, \ldots, R\}$, where $\hat{v}_{ijt,t+1}(r)$ is the slope of $\widehat{V}_{ijt,t+1}(\cdot)$ over $(r-1, r)$. That is, $\hat{v}_{ijt,t+1}(r) = \widehat{V}_{ijt,t+1}(r) - \widehat{V}_{ijt,t+1}(r-1)$. Then, problem (18) can be written as

$$\max_{x_t, f_{t+1}, z_{t,t+1}} \quad \sum_{i,j \in \mathcal{L}} \sum_{k \in \mathcal{K}} c_{ijkt}\ x_{ijkt} + \sum_{i,j \in \mathcal{L}} \sum_{r=1}^{R} \hat{v}_{ijt,t+1}(r)\ z_{ijt,t+1}(r) \tag{19}$$

$$\begin{aligned} \text{subject to} \quad & (9),\ (10),\ (11),\ (12) & & \\ & f_{ijt,t+1} - \sum_{r=1}^{R} z_{ijt,t+1}(r) = 0 & & \text{for all } i, j \in \mathcal{L} \qquad (20) \\ & 0 \le z_{ijt,t+1}(r) \le 1 & & \text{for all } i, j \in \mathcal{L},\ r = 1, \ldots, R. \end{aligned}$$

Representing a separable, piecewise-linear, concave objective function in a min-cost (max-contribution) network flow problem as in Figure 1, problem (19) is the min-cost network flow problem shown in Figure 2. Constraints (9), (12) and (20) are respectively the flow balance constraints for the nodes on the left, middle and right side of the figure. Since the problem is a min-cost network flow problem, integrality constraints (11) can be replaced by nonnegativity constraints.

We note that $f_{ijt,t+1}$ is the number of vehicles dispatched from location i to j at time period t. Therefore, $\widehat{V}_{ijt,t+1}(f)$ can be interpreted as the approximation to the expected future benefit from dispatching f vehicles from location i to j at time period t. This interpretation will be useful in updating and improving the value function approximations.

The remainder of this section describes two possible probability laws for the random variable A_t.

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

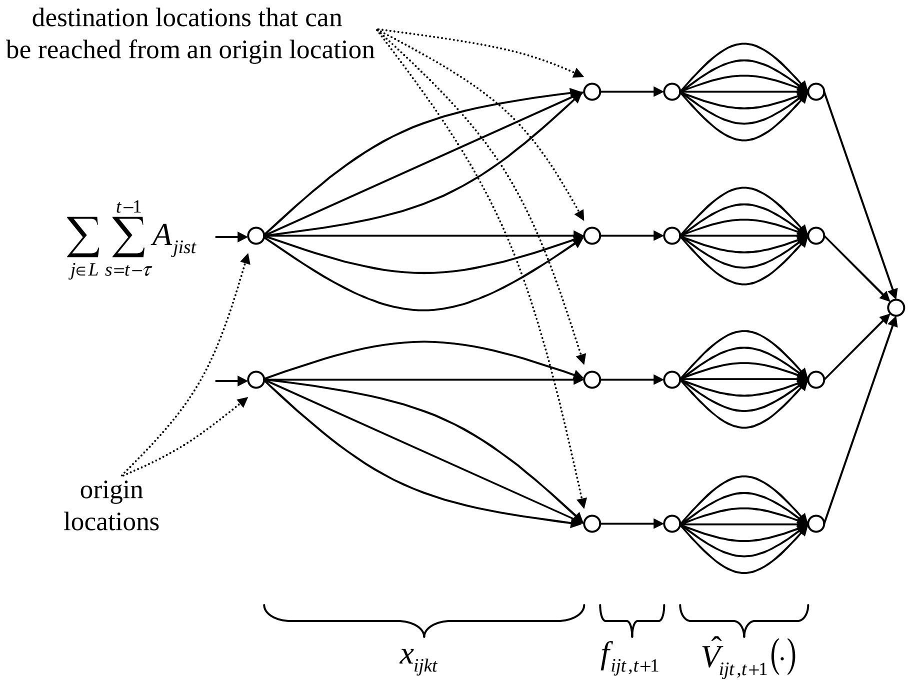

destination locations that can be reached from an origin location
$\sum_{j\in L}\sum_{s=t-\tau}^{t-1} A_{jist}$
origin locations
x_{ijkt}
$f_{ijt,t+1}$
$\hat{V}_{ijt,t+1}(\cdot)$

Figure 2: Problem (19) with separable, piecewise-linear, concave value function approximations is a min-cost network flow problem.

has not reached location j before time period t, the probability that this vehicle reaches location j at time period t is

$$\mathbb{P}\{\tau_{ij} = t - s \mid \tau_{ij} \geq t - s\}.$$

There are f_{ijst} vehicles that were dispatched from location i to j at time period s and that have not reached location j before time period t. Then, the number of vehicles that were dispatched from location i to j at time period s and that arrive at location j at time period t is binomially distributed with parameters f_{ijst} and $\mathbb{P}\{\tau_{ij} = t - s \mid \tau_{ij} \geq t - s\}$. That is, we have

$$\mathbb{P}\{A_{ijst} = a \mid f_{ijst} = f\} = \binom{f}{a} \Big[\mathbb{P}\{\tau_{ij} = t - s \mid \tau_{ij} \geq t - s\}\Big]^a \Big[\mathbb{P}\{\tau_{ij} > t - s \mid \tau_{ij} \geq t - s\}\Big]^{f-a}$$

for all $i, j \in \mathcal{L}$, $s = t - \tau, \ldots, t - 1$. This model is applicable when the travel times depend on conditions internal to the vehicles or drivers, such as breakdowns and skill levels.

Dependent Model – In this model, all vehicles dispatched from location i to j at time period s arrive at location j at the same time period. There are f_{ijst} vehicles that were dispatched from location i to j at time period s and that have not reached location j before time period t. Then, under the dependent model, either all of these vehicles reach location j at time period t (this happens with probability $\mathbb{P}\{\tau_{ij} = t - s \mid \tau_{ij} \geq t - s\}$) or none of these vehicles reach location j at time period

t (this happens with probability $\mathbb{P}\{\tau_{ij} > t - s \mid \tau_{ij} \geq t - s\}$). Therefore, we have

$$\mathbb{P}\{A_{ijst} = a \mid f_{ijst} = f\} = \begin{cases} 1 & \text{if } a = 0,\ f = 0 \\ \mathbb{P}\{\tau_{ij} = t - s \mid \tau_{ij} \geq t - s\} & \text{if } a = f,\ f > 0 \\ \mathbb{P}\{\tau_{ij} > t - s \mid \tau_{ij} \geq t - s\} & \text{if } a = 0,\ f > 0 \\ 0 & \text{otherwise} \end{cases}$$

for all $i, j \in \mathcal{L}$, $s = t - \tau, \ldots, t - 1$. This model is applicable when the travel times depend on external conditions, such as weather and traffic.

We note that for both of these arrival models, if a vehicle is dispatched from location i to j at time period s, then it arrives at location j 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 the vehicles dispatched from location i to j at a particular time period.

These two models represent two extremes and the travel times may have much more complex dependencies in practice. For example, we would expect the travel times for the vehicles leaving (or arriving at) the same or nearby locations at the same time period to be strongly correlated, if the travel times are affected by external conditions, such as weather or traffic.

4 Solution Method

The quality of the solutions obtained by solving approximate subproblems of form (18) depends on how "well" the value function approximations "mimic" the exact value function. In general, it is hard to come up with a high quality approximation, and therefore, we use a sampling-based method that iteratively improves the quality of the value function approximations.

Denoting the value function approximations at iteration n by $\{\widehat{V}_t^n(\cdot) : t \in \mathcal{T}\}$, we simulate the system by sampling a realization of A_t and D_t, say $\widehat{A}_t^n$ and $\widehat{D}_t^n$, and solving an approximate subproblem of the form

$$(x_t^n, f_{t+1}^n) = \underset{(x_t, f_{t+1}) \in \mathcal{Y}(f_t^n, \widehat{A}_t^n, \widehat{D}_t^n)}{\arg\max} \quad c_t x_t + \widehat{V}_{t+1}^n(f_{t+1}) \tag{21}$$

for each time period $t \in \mathcal{T}$. Note that the state vector f_{t+1}^n used at time period $t + 1$ is obtained from the solution of the approximate subproblem for time period t. After having solved (21) for all $t \in \mathcal{T}$, the challenge is to use the solutions of these subproblems to improve the approximations $\{\widehat{V}_t^n(\cdot) : t \in \mathcal{T}\}$. The methodology that we use for this purpose is based on the following four observations.

Observation 1 – In order to make the decisions for time period t, we only need the value function approximation components $\{\widehat{V}^n_{ijt,t+1}(\cdot) : i, j \in \mathcal{L}\}$ (see problem (18)). There is no need to improve the value function approximation components $\{\widehat{V}^n_{ijs,t+1}(\cdot) : i, j \in \mathcal{L},\ s = t+1-\tau, \ldots, t-1\}$.

Observation 2 – $\widehat{V}^n_{ijt,t+1}(f)$ approximates the expected future benefit from dispatching f vehicles from location i to j at time period t. At iteration n, $f^n_{ijt,t+1}$ vehicles are dispatched from location i to j at time period t. Therefore, $\hat{v}^n_{ijt,t+1}(f^n_{ijt,t+1}+1) = \widehat{V}^n_{ijt,t+1}(f^n_{ijt,t+1}+1) - \widehat{V}^n_{ijt,t+1}(f^n_{ijt,t+1})$ approximates the expected benefit from dispatching an additional vehicle from location i to j at time period t.

Observation 3 – The approximate subproblem solved for time period t is

$$\begin{aligned} \widetilde{V}^n_t(f^n_t, \widehat{A}^n_t, \widehat{D}^n_t) = \max_{x_t, f_{t+1}} \quad & \sum_{i,j\in\mathcal{L}} \sum_{k\in\mathcal{K}} c_{ijkt}\ x_{ijkt} + \widehat{V}^n_{ijt,t+1}(f_{ijt,t+1}) \\ \text{subject to} \quad & \sum_{j\in\mathcal{L}} \sum_{k\in\mathcal{K}} x_{ijkt} = \sum_{j\in\mathcal{L}} \sum_{s=t-\tau}^{t-1} \widehat{A}^n_{jist} \qquad \text{for all } i \in \mathcal{L} \qquad (22) \\ & (10),\ (11),\ (12). \end{aligned}$$

The right side of constraints (22) is the number of vehicles available at different locations at time period t. Let $\{\xi^n_{it} : i \in \mathcal{L}\}$ be the optimal values of the dual variables associated with these constraints at iteration n. Then, ξ^n_{it} gives an estimate of the expected benefit from having an additional vehicle at location i at time period t.

Observation 4 – Under the two arrival models discussed in Section 3.2, an additional vehicle dispatched from location i to j at time period t will arrive at location j at time period u with probability $\mathbb{P}\{\tau_{ij} = u - t\}$. Therefore, in view of Observation 3, at iteration n,

$$\vartheta^n_{ijt} = \sum_{u=t+1}^{(t+\tau)\wedge T} \mathbb{P}\{\tau_{ij} = u - t\}\ \xi^n_{ju}$$

gives an estimate of the expected benefit from dispatching an additional vehicle from location i to j at time period t. (We use $a \wedge b = \min\{a, b\}$.)

To sum up, at iteration n, we dispatch $f^n_{ijt,t+1}$ vehicles from location i to j at time period t and $\hat{v}^n_{ijt,t+1}(f^n_{ijt,t+1}+1)$ approximates the expected benefit from an additional vehicle dispatched from location i to j at time period t (Observation 2). Through the solution of the approximate subproblems at iteration n, we estimate the same quantity by ϑ^n_{ijt} (Observations 3 and 4). We use this new information to improve the value function approximation $\widehat{V}^n_{ijt,t+1}(\cdot)$ by using the following smoothing operation.

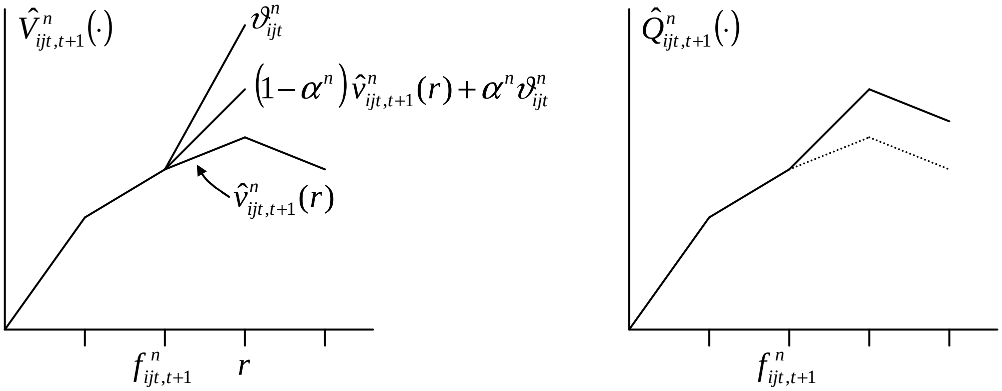


Figure 3: Updating the value function approximation $\widehat{V}^n_{ijt,t+1}(\cdot)$ using ϑ^n_{ijt}. The function $\widehat{Q}^n_{ijt,t+1}(\cdot)$ obtained after local smoothing may not be concave.

Step 1 – Set

$$q^n_{ijt,t+1}(r) = \begin{cases} (1-\alpha^n)\ \hat{v}^n_{ijt,t+1}(r) + \alpha^n\ \vartheta^n_{ijt} & \text{for } r = f^n_{ijt,t+1} + 1 \\ \hat{v}^n_{ijt,t+1}(r) & \text{for } r = 1, \ldots, f^n_{ijt,t+1}, (f_{ijt,t+1}+2), \ldots, R, \end{cases}$$

where $\alpha^n \in (0,1)$ is the step size at iteration n.

Step 2 – Set the vector $\hat{v}^{n+1}_{ijt,t+1} = \{\hat{v}^{n+1}_{ijt,t+1}(r) : r = 1, \ldots, R\}$, which characterizes the value function approximation $\widehat{V}^{n+1}_{ijt,t+1}(\cdot)$ at the next iteration, to

$$\hat{v}^{n+1}_{ijt,t+1} = \arg\min_{z} \quad \sum_{r=1}^{R} \left(z(r) - q^n_{ijt,t+1}(r)\right)^2 \tag{23}$$

$$\text{subject to} \quad z(r) - z(r-1) \leq 0 \qquad \text{for all } r = 2, \ldots, R. \tag{24}$$

In Step 1, we smooth the slope of $\widehat{V}^n_{ijt,t+1}(\cdot)$ at the relevant point by using the new information. However, after this smoothing, the function $Q^n_{ijt,t+1}(\cdot)$ characterized by the sequence of slopes $\{q^n_{ijt,t+1}(r) : r = 1, \ldots, R\}$ is not necessarily concave (see Figure 3). In Step 2, we find the "closest" concave function to $Q^n_{ijt,t+1}(\cdot)$ in the sense of objective function (23). Constraints (24) ensure that $\widehat{V}^{n+1}_{ijt,t+1}(\cdot)$ is concave. This updating method is due to Powell, Ruszczynski and Topaloglu (2004), where the authors show that there is a simple expression for the optimal solution to problem (23).

Figure 4 describes our solution methodology. Note that we obtain a sample realization of A_t just before solving the approximate subproblem for time period t, and hence, assume that the travel times are not known until the arrival actually occurs (that is, until the trip is completed).

Step 1 – Initialize iteration counter $n = 1$. Initialize $\{\widehat{V}^n_{ijt,t+1}(\cdot) : i, j \in \mathcal{L},\ t \in \mathcal{T}\}$ to arbitrary piecewise-linear, concave functions with points of nondifferentiability being a subset of integers.

Step 2 – Initialize time counter $t = 1$. Initialize f^n_1 to reflect the initial state of the vehicles.

Step 3 – Given f^n_t, sample a realization of A_t and D_t, say $\widehat{A}^n_t$ and $\widehat{D}^n_t$.

Step 4 – Solve approximate subproblem (18) for the current time period. Set

$$(x^n_t, f^n_{t+1}) = \arg\max_{x_t, f_{t+1}} \quad \sum_{i,j\in\mathcal{L}} \sum_{k\in\mathcal{K}} c_{ijkt}\ x_{ijkt} + \sum_{i,j\in\mathcal{L}} \widehat{V}_{ijt,t+1}(f_{ijt,t+1})$$

$$\begin{array}{lll} \text{subject to} & \displaystyle\sum_{j\in\mathcal{L}} \sum_{k\in\mathcal{K}} x_{ijkt} = \sum_{j\in\mathcal{L}} \sum_{s=t-\tau}^{t-1} \widehat{A}^n_{jist} & \text{for all } i \in \mathcal{L} \qquad (25) \\ & x_{ijkt} \leq \widehat{D}^n_{ijkt} & \text{for all } i, j \in \mathcal{L},\ k \in \mathcal{K} \\ & f_{ijs,t+1} = f^n_{ijst} - \widehat{A}^n_{ijst} & \text{for all } i, j \in \mathcal{L},\ s = t+1-\tau, \ldots, t-1 \\ & (11),\ (12). & \end{array}$$

Let $\{\xi^n_{it} : i \in \mathcal{L}\}$ be the optimal values of the dual variables associated with constraints (25).

Step 5 – Set $t = t + 1$. If $t \leq T$, go to Step 3.

Step 6 – Update the value function approximations for all $i, j \in \mathcal{L}$, $t \in \mathcal{T}$. Set

$$\vartheta^n_{ijt} = \sum_{u=t+1}^{(t+\tau)\wedge T} \mathbb{P}\{\tau_{ij} = u - t\}\ \xi^n_{ju}$$

$$q^n_{ijt,t+1}(r) = \begin{cases} (1-\alpha^n)\ \hat{v}^n_{ijt,t+1}(r) + \alpha^n\ \vartheta^n_{ijt} & \text{for } r = f^n_{ijt,t+1} + 1 \\ \hat{v}^n_{ijt,t+1}(r) & \text{otherwise} \end{cases}$$

$$\hat{v}^{n+1}_{ijt,t+1} = \arg\min_z \quad \sum_{r=1}^{R} \big(z(r) - q^n_{ijt,t+1}(r)\big)^2$$

$$\text{subject to} \quad z(r) - z(r-1) \leq 0 \qquad \text{for all } r = 2, \ldots, R.$$

Step 7 – Set $n = n + 1$. If one more iteration is needed, go to Step 2.

Figure 4: The general solution methodology.

5 Parallelization Issues

It is easy to see that approximate subproblem (19) decomposes to $|\mathcal{L}|$ subproblems, where, combining constraints (12) and (20) into

$$\sum_{k\in\mathcal{K}} x_{ijkt} - \sum_{r=1}^{R} z_{ijt,t+1}(r) = 0 \qquad \text{for all } i,j\in\mathcal{L},$$

the subproblem for location $i\in\mathcal{L}$ has the form

$$\max_{x_{it}, z_{it,t+1}} \quad \sum_{j\in\mathcal{L}}\sum_{k\in\mathcal{K}} c_{ijkt}\; x_{ijkt} + \sum_{j\in\mathcal{L}}\sum_{r=1}^{R} \hat{v}_{ijt,t+1}(r)\; z_{ijt,t+1}(r) \tag{26}$$

$$\begin{aligned}
\text{subject to} \quad & \sum_{j\in\mathcal{L}}\sum_{k\in\mathcal{K}} x_{ijkt} = \sum_{j\in\mathcal{L}}\sum_{s=t-\tau}^{t-1} A_{jist} \\
& \sum_{k\in\mathcal{K}} x_{ijkt} - \sum_{r=1}^{R} z_{ijt,t+1}(r) = 0 && \text{for all } j\in\mathcal{L} \\
& 0 \le x_{ijkt} \le D_{ijkt} && \text{for all } j\in\mathcal{L},\ k\in\mathcal{K} \\
& 0 \le z_{ijt,t+1}(r) \le 1 && \text{for all } j\in\mathcal{L},\ r=1,\ldots,R.
\end{aligned}$$

The fact that the problems of the form above corresponding to different $i\in\mathcal{L}$ can be solved in parallel gives our model a dramatic runtime advantage. Furthermore, as we shortly show, problem (26) can be solved by a simple sort operation. In order to show this, without loss of generality, we assume that $c_{ijKt} \ge c_{ij,K-1,t} \ge \ldots \ge c_{ij1t} \ge c_{ij0t}$ for all $i,j\in\mathcal{L}$, $t\in\mathcal{T}$. This implies that a load from location i to j and corresponding to movement mode k is served only if all the loads from location i to j and corresponding to movement modes $k+1,\ldots,K$ are served. We let

$$\hat{c}_{ijt}(r) = \begin{cases}
c_{ijKt} & \text{for } r = 1,\ldots,D_{ijKt} \\
c_{ij,K-1,t} & \text{for } r = (D_{ijKt}+1),\ldots,(D_{ij,K-1,t}+D_{ijKt}) \\
\quad\vdots & \\
c_{ijkt} & \text{for } r = (D_{ij,k+1,t}+\ldots+D_{ijKt}+1),\ldots,(D_{ijkt}+\ldots+D_{ijKt}) \\
\quad\vdots & \\
c_{ij0t} & \text{for } r = (D_{ij1t}+\ldots+D_{ijKt}+1),\ldots,(D_{ij0t}+\ldots+D_{ijKt}),
\end{cases}$$

for all $j\in\mathcal{L}$. Therefore, the vector $\{\hat{c}_{ijt}(r) : r = 1,\ldots,R\}$ gives a list of the most profitable R loads that need to be carried from location i to j, ranked according to their profitability. (Since $D_{ij0t} = \infty$ by definition, the vector $\{\hat{c}_{ijt}(r) : r = 1,\ldots,(D_{ij0t}+\ldots+D_{ijKt})\}$ has more than R elements.) Consider the problem

$$\max_{y_{it}} \quad \sum_{j\in\mathcal{L}}\sum_{r=1}^{R} \left[\hat{c}_{ijt}(r) + \hat{v}_{ijt,t+1}(r)\right] y_{ijt}(r) \tag{27}$$

$$\begin{aligned}
\text{subject to} \quad & \sum_{j\in\mathcal{L}}\sum_{r=1}^{R} y_{ijt}(r) = \sum_{j\in\mathcal{L}}\sum_{s=t-\tau}^{t-1} A_{jist} \\
& 0 \le y_{ijt}(r) \le 1 && \text{for all } j\in\mathcal{L},\ r=1,\ldots,R.
\end{aligned}$$

Problem (27) is a 0-1 knapsack problem, where each item consumes one unit of space. Therefore, it can be solved by sorting the objective function coefficients of the decision variables. The next proposition, which we prove in the appendix, shows that an optimal solution to problem (26) can be constructed by using an optimal solution to problem (27), and thereby, establishes the fact that problem (26) can be solved by a sort operation.

Proposition 1 *By using an optimal solution to problem* (27)*, one can construct an optimal solution to problem* (26)*.*

Topaloglu and Powell (to appear) use a model similar to that in Section 3.1 for fleet management problems with heterogeneous fleets. They introduce an additional superscript in their decision and state variables describing the vehicle type. For example, x^v_{ijkt} becomes the number of vehicles of type v dispatched from location i to j at time period t using movement mode k. Then, constraints (2) have to be written as

$$\sum_{v \in \mathcal{V}} x^v_{ijkt} \leq D_{ijkt} \qquad \qquad \text{for all } i, j \in \mathcal{L},\ k \in \mathcal{K},$$

where $\mathcal{V}$ is the set of vehicle types. Topaloglu and Powell (to appear) show that if there are multiple vehicle types, then subproblem (7) with separable, piecewise-linear, concave approximations becomes a *min-cost integer multi-commodity network flow problem*, for which it may be hard to obtain integer solutions. Although we do not prove here, it can be shown that subproblem (26) is a min-cost network flow problem even in the presence of multiple vehicle types.

6 Numerical Experiments

In this section, we test our model on the following four problem classes.

1) Problems with deterministic load arrivals and deterministic travel times – These problems can be formulated as min-cost network flow problems using a “state-time network” formulation.

2) Problems with random load arrivals and deterministic travel times – The model in Section 3.1 applies to these problems.

3) Problems with deterministic load arrivals and random travel times – For this and the next class of problems, we present comparisons with typical engineering practices, which assume that the realization of the travel times will be equal to their expected values.

4) Problems with random load arrivals and random travel times.

We include problems with deterministic travel times in our numerical experiments because there

Problem	T	$\lvert\mathcal{L}\rvert$	$\bar{\tau}$	R	$\bar{D}$	c_0
Base	60	40	3.64	200	2500	4
T 30	30	40	3.64	200	1250	4
T 90	90	40	3.64	200	3750	4
$\mathcal{L}$ 20	60	20	3.64	200	2500	4
$\mathcal{L}$ 60	60	60	3.64	200	2500	4
$\bar{\tau}$ 2.60	60	40	2.60	200	2500	4
$\bar{\tau}$ 5.33	60	40	5.33	200	2500	4
R 100	60	40	3.64	100	2500	4
R 400	60	40	3.64	400	2500	4
R 600	60	40	3.64	600	2500	4
c_0 2	60	40	3.64	200	2500	2
c_0 8	60	40	3.64	200	2500	8

Table 1: Characteristics of the test problems used in our experiments.

exist a variety of methods to obtain optimal or high-quality solutions for them. This gives us a chance to carefully test the performance of our model. Also, our model may still be preferable under deterministic travel times because of its decomposed decision-making structure and fast runtime.

In our experimental setup, we generate one basic problem and modify its certain attributes to obtain different problems. Table 1 describes these problems. In this table, T is the length of the planning horizon, $\mathcal{L}$ is the number of locations, $\bar{\tau}$ is the average of the expected travel time over all origin-destination pairs, R is the number of available vehicles, $\bar{D}$ is the expected number of loads over the whole planning horizon and c_0 is the empty repositioning cost applied on "per mile" basis. We generate problems such that the number of loads outbound from a location is negatively correlated with the number of loads inbound to a location (see Godfrey and Powell (2002a)). We expect these problems to require more repositioning moves in the optimal solution, and hence, to be more difficult.

6.1 Problems with Deterministic Load Arrivals and Deterministic Travel Times

Assuming that D_{ijkt} and τ_{ij} are deterministic for all $i, j \in \mathcal{L},\ k \in \mathcal{K},\ t \in \mathcal{T}$, the optimal objective value can be found by using a "state-time network" model that amounts to solving

$$\begin{aligned}
\max_x \quad & \sum_{t\in\mathcal{T}} \sum_{i,j\in\mathcal{L}} \sum_{k\in\mathcal{K}} c_{ijkt}\ x_{ijkt} && && (28)\\
\text{subject to} \quad & -\sum_{\substack{j\in\mathcal{L}:\\ t-\tau_{ji}\geq 1}} \sum_{k\in\mathcal{K}} x_{jik,t-\tau_{ji}} + \sum_{j\in\mathcal{L}} \sum_{k\in\mathcal{K}} x_{ijkt} = \sum_{\substack{j\in\mathcal{L}:\\ t-\tau_{ji}\leq 0}} f_{ji,t-\tau_{ji},1} && \text{for all } i \in \mathcal{L},\ t \in \mathcal{T} &&\\
& (2),\ (3) && \text{for all } i, j \in \mathcal{L},\ k \in \mathcal{K},\ t \in \mathcal{T}, &&
\end{aligned}$$

where $f_1 = \{f_{ijs1} : i, j \in \mathcal{L},\ s = 1-\tau, \ldots, 0\}$ is a part of the problem data. The first set of constraints state that the number of vehicles that reach location i at time period t is equal to the number of vehicles that depart from location i at time period t. Their right side is the number of vehicles that reach location i at time period t and that were dispatched before the planning horizon.

In order to test our approach, we apply the algorithm in Figure 4 for 250 iterations and compare

	Perc. of opt. obj.			Secs.
Pr.	50	100	250	/itn.
Base	98.63	99.22	99.26	0.9
T 30	98.47	99.13	99.21	0.4
T 90	98.09	98.95	99.07	1.4
$\mathcal{L}$ 20	99.14	99.45	99.60	0.2
$\mathcal{L}$ 60	97.08	98.94	99.04	2.1
$\bar{\tau}$ 2.60	97.62	99.35	99.52	1.0
$\bar{\tau}$ 5.33	96.84	98.09	98.31	0.8
R 100	96.90	97.71	98.35	0.8
R 400	97.84	99.53	99.75	1.1
R 600	97.30	99.61	99.77	1.1
c_0 2	96.68	99.10	99.26	0.9
c_0 8	98.89	99.38	99.48	0.9

Table 2: Results for problems with deterministic load arrivals and deterministic travel times.

the objective value at each iteration with the objective value of problem (28). In Table 2, we show the objective value provided by our approach after 50, 100 and 250 iterations, and the runtime per iteration. (An iteration involves solving T approximate subproblems of form (18) and updating the value function approximations.) For a majority of problem instances, it takes less than 100 iterations for our approach to provide solutions within 1% of the optimal objective value.

6.2 Problems with Random Load Arrivals and Deterministic Travel Times

For this problem class, we use two suboptimal benchmark strategies. The first one is the model presented in Section 3.1. Topaloglu and Powell (to appear) report that this model yields high-quality solutions. The second one is a common engineering practice referred to as the *rolling horizon method.* This strategy solves a deterministic, multi-stage problem using the actual load realizations for the current time period and the expectations of the load random variables for the future time periods. In particular, for a given system state $f_t = \{f_{ijst} : i, j \in \mathcal{L},\ s = t - \tau, \ldots, t-1\}$ and load realization D_t at time period t, the N-period rolling horizon method solves

$$\max_x \quad \sum_{u=t}^{t+N} \sum_{i,j \in \mathcal{L}} \sum_{k \in \mathcal{K}} c_{ijku}\, x_{ijku} \tag{29}$$

$$\text{subject to} \quad -\sum_{\substack{j \in \mathcal{L}: \\ u-\tau_{ji} \geq t}} \sum_{k \in \mathcal{K}} x_{jik,u-\tau_{ji}} + \sum_{j \in \mathcal{L}} \sum_{k \in \mathcal{K}} x_{ijku} = \sum_{\substack{j \in \mathcal{L}: \\ u-\tau_{ji} \leq t-1}} f_{ji,u-\tau_{ji},t} \qquad \text{for all } i \in \mathcal{L},\ u = t, \ldots, t+N \tag{30}$$

$$x_{ijkt} \leq D_{ijkt} \qquad \text{for all } i, j \in \mathcal{L},\ k \in \mathcal{K}, \tag{31}$$

$$x_{ijku} \leq \mathbb{E}\{D_{ijku}\} \qquad \text{for all } i, j \in \mathcal{L},\ k \in \mathcal{K},\ u = t+1, \ldots, t+N \tag{32}$$

$$x_{ijku} \in \mathbb{Z}_+ \qquad \text{for all } i, j \in \mathcal{L},\ k \in \mathcal{K},\ u = t, \ldots, t+N \tag{33}$$

in order to make the decisions at time period t. Problem (29) is similar to problem (28), but it includes decision variables only for the time periods $t, \ldots, t+N$. Although this problem includes

N	1	5	10	15	20
Avg. obj. (millions)	3.95	10.26	10.51	10.53	10.52

Table 3: Performance of the rolling horizon method on problem Base with different values of N.

decision variables for the time periods $t, \ldots, t+N$, we only implement the decisions corresponding to time period t and re-solve a similar problem when making the decisions for time period $t+1$. To find an appropriate value for N, we applied the rolling horizon method on problem Base with different values of N for 500 samples of $\{D_t : t \in \mathcal{T}\}$. Table 3 shows that the average objective value over 500 samples stabilizes when N is around 10, 15. To be on the safe side, we use $N = 15$.

In a stochastic setting, testing the models presented in Sections 3.1 and 3.2 involves two sets of iterations. In the *training iterations*, we essentially follow the algorithm in Figure 4 by sampling a load realization and solving the approximate subproblem (18) (or subproblem (7) if we are using the model in Section 3.1) for all time periods, and updating the value function approximations. In the *testing iterations*, we do not update the value function approximations anymore and only test the solution quality provided by the approximations obtained during the training iterations. We use 50, 250 and 500 training iterations to build three sets of value function approximations and we evaluate their performances using 500 testing iterations. (To reduce the sampling noise, we use the same load realizations to evaluate the performance of each set of value function approximations.)

The results are presented in Table 4. DeT, RaT and RoH respectively refers to the deterministic and random travel time models of Sections 3.1 and 3.2, and the rolling horizon method. The first set of columns show the average objective value over 500 testing iterations after training the value function approximations for 50, 250 and 500 iterations. The second set of columns show the percent deviation of the objective value of DeT and RoH from that of RaT, indicating the value of using RaT over DeT or RoH. For each testing iteration, we also give each method a rank depending on the objective value it yields (1 being the best performing). In the third set of columns, we show the average ranking of each method over 500 testing iterations.

After 500 training iterations, DeT performs better than RaT, and RaT performs better than RoH in terms of average objective value. Furthermore, the average rankings point out that the objective values provided by DeT are better than those provided by RaT in all testing iterations. This discrepancy in the performances of RaT and DeT can be explained as follows. For simplicity, we assume that all travel times are one time period. Under this assumption, we have $\tau = 1$, $r_t = \{r_{itt} : i \in \mathcal{L}\}$, $f_t = \{f_{ij,t-1,t} : i, j \in \mathcal{L}\}$ and $r_{itt} = \sum_{j \in \mathcal{L}} f_{ji,t-1,t}$. DeT uses approximations of the

		Avg. obj. (millions)			Perc. dev. fr. RaT			Avg. rank			Secs.
Pr.	Alg.	50	250	500	50	250	500	50	250	500	/itn.
Base	RaT	9.87	10.74	10.86				3.0	2.1	2.0	1.0
	DeT	11.12	11.43	11.49	12.6	6.5	5.8	1.0	1.0	1.0	1.5
	RoH	10.53	10.53	10.53	6.6	-2.0	-3.0	2.0	2.9	3.0	7.2
T 30	RaT	5.33	5.59	5.65				2.9	2.1	2.0	0.5
	DeT	5.80	5.94	5.96	8.8	6.3	5.5	1.0	1.0	1.0	0.7
	RoH	5.45	5.45	5.45	2.2	-2.6	-3.6	2.1	2.9	3.0	2.8
T 90	RaT	14.35	15.89	16.06				3.0	2.1	2.0	1.4
	DeT	16.45	16.92	16.99	14.7	6.4	5.8	1.0	1.0	1.0	2.4
	RoH	15.62	15.62	15.62	8.9	-1.7	-2.7	2.0	2.9	3.0	10.2
$\mathcal{L}$ 20	RaT	11.69	12.29	12.37				3.0	2.1	2.0	0.3
	DeT	12.13	12.44	12.53	3.8	1.3	1.3	1.6	1.1	1.0	0.6
	RoH	12.15	12.15	12.15	4.0	-1.1	-1.8	1.4	2.8	3.0	1.4
$\mathcal{L}$ 60	RaT	9.20	10.29	10.38				3.0	2.0	2.0	2.2
	DeT	10.94	11.25	11.32	18.9	9.4	9.0	1.0	1.0	1.0	3.0
	RoH	9.87	9.87	9.87	7.3	-4.0	-4.9	2.0	3.0	3.0	17.7
$\bar{\tau}$ 2.60	RaT	10.54	11.45	11.60				3.0	2.5	2.2	1.0
	DeT	12.27	12.47	12.49	16.4	8.9	7.6	1.0	1.0	1.0	1.1
	RoH	11.47	11.47	11.47	8.8	0.2	-1.1	2.0	2.5	2.8	7.4
$\bar{\tau}$ 5.33	RaT	8.84	9.39	9.44				2.7	2.0	2.0	0.9
	DeT	9.20	9.69	9.77	4.1	3.2	3.5	1.0	1.0	1.0	2.0
	RoH	8.92	8.92	8.92	1.0	-4.9	-5.5	2.3	3.0	3.0	6.1
R 100	RaT	6.60	7.09	7.15				2.8	2.0	2.0	0.8
	DeT	7.16	7.47	7.51	8.5	5.4	5.1	1.0	1.0	1.0	1.1
	RoH	6.69	6.69	6.69	1.4	-5.6	-6.4	2.2	3.0	3.0	6.1
R 400	RaT	11.81	12.60	12.95				3.0	2.8	2.0	1.1
	DeT	13.14	13.33	13.40	11.3	5.8	3.5	1.0	1.0	1.0	1.7
	RoH	12.78	12.78	12.78	8.2	1.5	-1.3	2.0	2.2	3.0	5.8
R 600	RaT	12.42	13.27	13.65				3.0	2.6	2.0	1.0
	DeT	13.60	13.79	13.84	9.5	3.9	1.4	1.0	1.0	1.0	1.6
	RoH	13.34	13.34	13.34	7.4	0.5	-2.3	2.0	2.4	3.0	5.8
c_0 2	RaT	10.81	11.75	11.86				3.0	2.1	2.0	0.9
	DeT	12.00	12.48	12.54	11.0	6.2	5.8	1.0	1.0	1.0	1.5
	RoH	11.54	11.54	11.54	6.7	-1.8	-2.7	2.0	2.9	3.0	8.3
c_0 8	RaT	8.79	9.37	9.48				3.0	2.1	2.0	0.9
	DeT	9.69	9.94	10.00	10.3	6.1	5.5	1.0	1.0	1.0	1.4
	RoH	9.14	9.14	9.14	4.0	-2.4	-3.6	2.0	2.9	3.0	5.7

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

form (see (8))

$$\widehat{V}_t(r_t) = \sum_{i \in \mathcal{L}} \widehat{V}_{itt}(r_{itt}) = \sum_{i \in \mathcal{L}} \widehat{V}_{itt}\left(\sum_{j \in \mathcal{L}} f_{ji,t-1,t}\right), \tag{34}$$

whereas RaT uses approximations of the form (see (16))

$$\widehat{V}_t(f_t) = \sum_{i,j \in \mathcal{L}} \widehat{V}_{ij,t-1,t}(f_{ij,t-1,t}). \tag{35}$$

The exact value function is a nonseparable function of $r_t = \{\sum_{j \in \mathcal{L}} f_{ji,t-1,t} : i \in \mathcal{L}\}$. (34) assumes that the value function is a separable function of $\{\sum_{j \in \mathcal{L}} f_{ji,t-1,t} : i \in \mathcal{L}\}$. (35) goes one step beyond by assuming that the value function is a separable function of $\{f_{ij,t-1,t} : i, j \in \mathcal{L}\}$, and hence, captures the characteristics of the exact value function less accurately. Nevertheless, as shown in the next two sections, RaT performs better than DeT when the travel times are random.

We finally note that the runtimes reported for RaT is for a serial implementation, where the subproblems corresponding to different locations are solved sequentially.

6.3 Problems with Deterministic Load Arrivals and Random Travel Times

In order to evaluate the performance of RaT under random travel times, we compare it with modified versions of DeT and RoH. When making the decisions for time period t, these modified versions assume that if a vehicle dispatched from location i to j at time period s has not yet reached its destination by time period t, then it will reach its destination at time period $s + \mathbb{E}\{\tau_{ij}\}$. In particular, for a given system state $f_t = \{f_{ijst} : i, j \in \mathcal{L},\ s = t - \tau, \ldots, t - 1\}$ and realization of arrivals $A_t = \{A_{ijst} : i, j \in \mathcal{L},\ s = t - \tau, \ldots, t - 1\}$ at time period t, DeT makes decisions by solving

$$\max_{x_t, r_{t+1}} \quad \sum_{i,j \in \mathcal{L}} \sum_{k \in \mathcal{K}} c_{ijkt}\ x_{ijkt} + \sum_{j \in \mathcal{L}} \sum_{u=t+1}^{t+\tau} \widehat{V}_{ju,t+1}(r_{ju,t+1}) \tag{36}$$

$$\begin{aligned} \text{subject to} \quad & \sum_{j \in \mathcal{L}} \sum_{k \in \mathcal{K}} x_{ijkt} = \sum_{j \in \mathcal{L}} \sum_{s=t-\tau}^{t-1} A_{jist} && \text{for all } i \in \mathcal{L} \\ & - \sum_{i \in \mathcal{L}} \sum_{k \in \mathcal{K}} \mathbf{1}_{\bar{\tau}_{ij}}(u - t)\ x_{ijkt} + r_{ju,t+1} = \sum_{\substack{i \in \mathcal{L}: \\ u - \bar{\tau}_{ij} \leq t-1}} \big(f_{ij,u-\bar{\tau}_{ij},t} - A_{ij,u-\bar{\tau}_{ij},t} \big) && \\ & && \text{for all } j \in \mathcal{L},\ u = t+1, \ldots, t+\tau \\ & (2),\ (3) && \text{for all } i, j \in \mathcal{L},\ k \in \mathcal{K}, \end{aligned}$$

where we denote $\mathbb{E}\{\tau_{ij}\}$ by $\bar{\tau}_{ij}$. This problem is similar to (7). Noting that $\sum_{j \in \mathcal{L}} \sum_{s=t-\tau}^{t-1} A_{jist}$ is the number of vehicles available at location i at time period t, the first set of constraints are analogous to constraints (1). The second set of constraints are analogous to constraints (4) that describe the system dynamics. The term $f_{ij,u-\bar{\tau}_{ij},t} - A_{ij,u-\bar{\tau}_{ij},t}$ on their right side is the number of vehicles that were dispatched from location i to j at time period $u - \bar{\tau}_{ij}$ and that have not reached location j by time period t. DeT assumes that these vehicles will reach location j at time period u $(= u - \bar{\tau}_{ij} + \bar{\tau}_{ij})$.

In order to make the decisions for time period t, RoH solves

$$\max_{x} \quad \sum_{u=t}^{t+N} \sum_{i,j \in \mathcal{L}} \sum_{k \in \mathcal{K}} c_{ijku}\ x_{ijku} \tag{37}$$

$$\begin{aligned} \text{subject to} \quad & \sum_{j \in \mathcal{L}} \sum_{k \in \mathcal{K}} x_{ijkt} = \sum_{j \in \mathcal{L}} \sum_{s=t-\tau}^{t-1} A_{jist} && \text{for all } i \in \mathcal{L} \\ & - \sum_{\substack{j \in \mathcal{L}: \\ u - \bar{\tau}_{ji} \geq t}} \sum_{k \in \mathcal{K}} x_{jik,u-\bar{\tau}_{ji}} + \sum_{j \in \mathcal{L}} \sum_{k \in \mathcal{K}} x_{ijku} = \sum_{\substack{j \in \mathcal{L}: \\ u - \bar{\tau}_{ji} \leq t-1}} \big(f_{ji,u-\bar{\tau}_{ji},t} - A_{ji,u-\bar{\tau}_{ji},t} \big) && \\ & && \text{for all } i \in \mathcal{L},\ u = t+1, \ldots, t+N \\ & (31), (32),\ (33). && \end{aligned}$$

Problem (37) is similar to problem (29). The first set of constraints state that the number of vehicles that depart from location i at time period t is equal to the number of vehicles that are available at location i at time period t. Since we solve problem (37) in order to make the decisions for time period t, the values of the random variables $\{A_{ijst} : i, j \in \mathcal{L},\ s = t - \tau, \ldots, t-1\}$ are known at this stage. The second set of constraints state that the number of vehicles that reach location i at time period u is equal to the number of vehicles that depart from location i at time period u. It can be shown that if τ_{ij} is deterministic and takes the value $\bar{\tau}_{ij}$, then problems (29) and (37) are the same. Clearly, DeT and RoH require $\mathbb{E}\{\tau_{ij}\}$ to be an integer, whereas this is not a requirement for RaT.

Table 5 shows that for problems with random travel times, RaT performs better than DeT and RoH even after 50 training iterations. By using a model that explicitly addresses random travel times (instead of simpler models that use only the expected values of the travel times), we can obtain about 5% increase in the average objective value.

6.4 Problems with Random Load Arrivals and Random Travel Times

Finally, we test our approach on problems with random load arrivals and random travel times. In this case, the benchmark models given in problems (36) and (37) continue to apply. Table 6 shows that after 250 training iterations RaT performs better than DeT and RoH. However, 50 training iterations may not be enough for RaT to provide better solutions than DeT. This indicates that RaT needs more training iterations to yield high-quality solutions in the presence of random load arrivals. Nevertheless, this does not make RaT less attractive for real-time applications since the value functions are trained in an offline setting.

7 Conclusion

In this paper, we presented a stochastic model for the dynamic fleet management problem with random travel times. On a variety of problem instances (with or without random travel times), we showed that our approach provides near-optimal or high-quality solutions quite fast.

In addition to handling random travel times, our model decomposes the fleet management problem by locations and solves one subproblem for each time period-location pair. This feature becomes especially attractive for real-world applications where there are multiple dispatchers responsible for making the vehicle allocation decisions at different locations, and the information available to each dispatcher includes only the number of inbound vehicles and outbound loads to/from its own location.

Several extensions to our model are possible. Heterogeneous fleets with multiple vehicle types can be handled by using the approach in Topaloglu and Powell (to appear). Our approach is simulation-based. Therefore, we can address load pickup time windows by simply carrying over the uncovered

		Avg. obj. (millions)			Perc. dev. fr. RaT			Avg. rank			Secs.
Pr.	Alg.	50	250	500	50	250	500	50	250	500	/itn.
Base	RaT	11.93	12.21	12.25				1.0	1.0	1.0	0.9
	DeT	11.63	11.68	11.69	-2.5	-4.4	-4.6	2.0	2.0	2.0	1.2
	RoH	11.36	11.36	11.36	-4.8	-7.0	-7.3	3.0	3.0	3.0	8.4
T 30	RaT	6.04	6.11	6.12				1.0	1.0	1.0	0.5
	DeT	5.71	5.74	5.74	-5.4	-6.1	-6.2	2.0	2.0	2.0	0.5
	RoH	5.51	5.51	5.51	-8.8	-9.9	-10.1	3.0	3.0	3.0	4.6
T 90	RaT	17.65	18.10	18.16				1.0	1.0	1.0	1.4
	DeT	16.99	17.06	17.07	-3.8	-5.8	-6.0	2.0	2.0	2.0	1.9
	RoH	16.63	16.63	16.63	-5.8	-8.1	-8.4	3.0	3.0	3.0	13.6
$\mathcal{L}$ 20	RaT	11.28	11.38	11.41				1.0	1.0	1.0	0.3
	DeT	10.77	10.81	10.83	-4.5	-5.1	-5.1	2.0	2.0	2.0	0.3
	RoH	10.49	10.49	10.49	-7.0	-7.8	-8.1	3.0	3.0	3.0	2.4
$\mathcal{L}$ 60	RaT	11.73	12.05	12.09				1.0	1.0	1.0	2.1
	DeT	11.52	11.55	11.56	-1.8	-4.2	-4.4	2.0	2.0	2.0	2.7
	RoH	11.10	11.10	11.10	-5.4	-7.9	-8.2	3.0	3.0	3.0	20.4
$\bar{\tau}$ 2.60	RaT	12.97	13.2	13.21				1.0	1.0	1.0	1.0
	DeT	12.50	12.52	12.52	-3.6	-5.2	-5.2	2.0	2.0	2.0	0.9
	RoH	12.06	12.06	12.06	-7.0	-8.7	-8.8	3.0	3.0	3.0	8.7
$\bar{\tau}$ 5.33	RaT	10.12	10.31	10.34				1.0	1.0	1.0	0.9
	DeT	9.69	9.76	9.78	-4.3	-5.4	-5.4	2.0	2.0	2.0	1.6
	RoH	9.48	9.48	9.48	-6.3	-8.0	-8.3	3.0	3.0	3.0	9.1
R 100	RaT	7.76	7.91	7.91				1.0	1.0	1.0	0.9
	DeT	7.47	7.49	7.50	-3.8	-5.3	-5.2	2.0	2.1	2.1	0.9
	RoH	7.30	7.30	7.30	-6.0	-7.7	-7.8	3.0	2.9	2.9	8.2
R 400	RaT	13.59	13.84	13.86				1.0	1.0	1.0	1.1
	DeT	13.14	13.19	13.19	-3.4	-4.7	-4.8	2.0	2.0	2.0	1.3
	RoH	12.80	12.8	12.80	-5.8	-7.5	-7.6	3.0	3.0	3.0	9.0
R 600	RaT	13.92	14.15	14.16				1.0	1.0	1.0	1.0
	DeT	13.54	13.58	13.59	-2.7	-4.0	-4.1	2.0	2.0	2.0	1.3
	RoH	13.19	13.19	13.19	-5.2	-6.8	-6.9	3.0	3.0	3.0	8.8
c_0 2	RaT	12.94	13.32	13.37				1.0	1.0	1.0	0.9
	DeT	12.57	12.63	12.63	-2.9	-5.2	-5.5	2.0	2.0	2.0	1.2
	RoH	12.38	12.38	12.38	-4.3	-7.0	-7.4	3.0	3.0	3.0	8.7
c_0 8	RaT	10.48	10.67	10.71				1.0	1.0	1.0	0.9
	DeT	10.11	10.17	10.17	-3.5	-4.7	-5.1	2.0	2.0	2.0	1.1
	RoH	9.94	9.94	9.94	-5.2	-6.9	-7.3	3.0	3.0	3.0	7.9

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

loads to the next time period. However, a thorough treatment requires including the loads in the state variable of the dynamic program and it is not yet clear how to approximate the value function when the state variable has a load dimension.

8 Appendix

In this section, we prove Proposition 1. We first show that by using an optimal solution to problem (27), we can construct a feasible solution to problem (26). For clarity, we assume that $\mathcal{K} = \{0, 1\}$. Generalizing the proof to arbitrary $\mathcal{K}$ is obvious.

		Avg. obj. (millions)			Perc. dev. fr. RaT			Avg. rank			Secs.
Pr.	Alg.	50	250	500	50	250	500	50	250	500	/itn.
Base	RaT	10.01	10.58	10.63				2.0	1.0	1.0	1.0
	DeT	10.25	10.39	10.38	2.4	-1.8	-2.4	1.1	2.0	2.0	1.2
	RoH	9.86	9.86	9.86	-1.5	-6.8	-7.2	2.9	3.0	3.0	9.3
T 30	RaT	5.36	5.58	5.63				1.7	1.1	1.0	0.5
	DeT	5.39	5.44	5.45	0.5	-2.6	-3.1	1.5	2.0	2.0	0.6
	RoH	5.16	5.16	5.16	-3.7	-7.6	-8.3	2.9	3.0	3.0	5.1
T 90	RaT	14.92	15.86	16.06				2.1	1.2	1.0	1.5
	DeT	15.32	15.45	15.48	2.7	-2.6	-3.6	1.1	1.9	2.0	2.0
	RoH	14.38	14.38	14.38	-3.6	-9.4	-10.5	2.9	3.0	3.0	14.5
$\mathcal{L}$ 20	RaT	10.68	11.14	11.12				1.3	1.0	1.0	0.3
	DeT	10.59	10.72	10.66	-0.8	-3.8	-4.1	1.8	2.0	2.0	0.4
	RoH	10.13	10.13	10.13	-5.1	-9.1	-8.9	3.0	3.0	3.0	2.4
$\mathcal{L}$ 60	RaT	9.23	9.97	10.14				2.1	1.2	1.1	2.2
	DeT	9.75	9.82	9.89	5.7	-1.5	-2.4	1.3	1.9	2.0	2.8
	RoH	9.15	9.15	9.15	-0.9	-8.2	-9.7	2.6	2.9	3.0	20.5
$\bar{\tau}$ 2.60	RaT	10.49	11.2	11.31				2.1	1.1	1.0	1.0
	DeT	10.77	10.9	10.87	2.7	-2.7	-3.9	1.1	2.0	2.0	0.9
	RoH	10.29	10.29	10.29	-1.9	-8.1	-9.0	2.9	3.0	3.0	8.6
$\bar{\tau}$ 5.33	RaT	8.86	9.18	9.22				1.9	1.1	1.0	0.9
	DeT	8.53	8.68	8.70	-3.8	-5.5	-5.7	1.3	1.9	2.0	1.6
	RoH	8.09	8.09	8.09	-8.7	-11.9	-12.3	2.8	3.0	3.0	10.1
R 100	RaT	6.71	6.96	7.01				1.6	1.1	1.0	0.9
	DeT	6.73	6.86	6.86	0.3	-1.4	-2.1	1.5	1.9	2.0	0.9
	RoH	6.35	6.35	6.35	-5.4	-8.8	-9.4	3.0	3.0	3.0	9.0
R 400	RaT	11.96	12.67	12.74				2.2	1.1	1.1	1.1
	DeT	12.32	12.50	12.45	3.0	-1.3	-2.3	1.0	1.9	2.0	1.4
	RoH	11.76	11.76	11.76	-1.7	-7.2	-7.8	2.8	3.0	3.0	9.0
R 600	RaT	12.23	12.96	13.06				2.4	1.3	1.1	1.2
	DeT	12.74	12.91	12.87	4.1	-0.4	-1.4	1.1	1.7	1.9	1.4
	RoH	12.27	12.27	12.27	0.3	-5.4	-6.0	2.5	3.0	3.0	11.5
c_0 2	RaT	10.74	11.34	11.41				1.9	1.0	1.0	1.0
	DeT	10.88	11.03	11.02	1.3	-2.8	-3.4	1.2	2.0	2.0	1.2
	RoH	10.52	10.52	10.52	-2.0	-7.2	-7.8	2.9	3.0	3.0	8.5
c_0 8	RaT	8.74	9.23	9.28				1.7	1.0	1.0	1.0
	DeT	8.81	8.96	8.95	0.8	-3.0	-3.6	1.4	2.0	2.0	1.1
	RoH	8.46	8.46	8.46	-3.2	-8.4	-8.9	2.9	3.0	3.0	8.6

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

Let $\{y^*_{ijt}(r) : j \in \mathcal{L},\ r = 1, \dots, R\}$ be an optimal solution to problem (27). Set

$$\tilde{x}_{ij1t} = \sum_{r=1}^{D_{ij1t}} y^*_{ijt}(r) \qquad \text{for all } j \in \mathcal{L} \tag{38}$$

$$\tilde{x}_{ij0t} = \sum_{r=D_{ij1t}+1}^{R} y^*_{ijt}(r) \qquad \text{for all } j \in \mathcal{L} \tag{39}$$

$$\tilde{z}_{ijt,t+1}(r) = y^*_{ijt}(r) \qquad \text{for all } j \in \mathcal{L},\ r = 1, \dots, R. \tag{40}$$

(Summations with the lower bound larger than the upper bound should be understood as 0.) By this construction, we have $\tilde{x}_{ij1t} \leq D_{ij1t}$ and $\tilde{x}_{ij0t} \leq D_{ij0t} = \infty$ and $z^*_{ijt,t+1}(r) \leq 1$ for all $j \in \mathcal{L},\ r = 1, \dots, R$. Since $\{y^*_{ijt}(r) : j \in \mathcal{L},\ r = 1, \dots, R\}$ is feasible to problem (27), (38)-(40) imply that

$$\sum_{j \in \mathcal{L}} \sum_{k=0}^{1} \tilde{x}_{ijkt} = \sum_{j \in \mathcal{L}} \sum_{r=1}^{R} y^*_{ijt}(r) = \sum_{j \in \mathcal{L}} \sum_{s=t-\tau}^{t-1} A_{jist}$$

$$\sum_{k=0}^{1} \tilde{x}_{ijkt} = \sum_{r=1}^{R} y^*_{ijt}(r) = \sum_{r=1}^{R} \tilde{z}_{ijt,t+1}(r) \qquad \text{for all } j \in \mathcal{L}.$$

Therefore, the solution constructed in (38)-(40) is feasible to problem (26). We note that the objective value of the solution constructed in (38)-(40) is equal to the objective value of the solution $\{y^*_{ijt}(r) : j \in \mathcal{L},\ r = 1, \ldots, R\}$ because

$$\begin{aligned}
&\sum_{j\in\mathcal{L}} \sum_{r=1}^{R} [\hat{c}_{ijt}(r) + \hat{v}_{ijt,t+1}(r)]\ y^*_{ijt}(r) \\
&\quad = \sum_{j\in\mathcal{L}} \sum_{r=1}^{D_{ij1t}} \hat{c}_{ijt}(r)\ y^*_{ijt}(r) + \sum_{j\in\mathcal{L}} \sum_{r=D_{ij1t}+1}^{R} \hat{c}_{ijt}(r)\ y^*_{ijt}(r) + \sum_{j\in\mathcal{L}} \sum_{r=1}^{R} \hat{v}_{ijt,t+1}(r)\ y^*_{ijt}(r) \\
&\quad = \sum_{j\in\mathcal{L}} c_{ij1t} \left[\sum_{r=1}^{D_{ij1t}} y^*_{ijt}(r) \right] + \sum_{j\in\mathcal{L}} c_{ij0t} \left[\sum_{r=D_{ij1t}+1}^{R} y^*_{ijt}(r) \right] + \sum_{j\in\mathcal{L}} \sum_{r=1}^{R} \hat{v}_{ijt,t+1}(r)\ \tilde{z}_{ijt,t+1}(r) \\
&\quad = \sum_{j\in\mathcal{L}} \sum_{k=0}^{1} c_{ijkt}\ \tilde{x}_{ijkt} + \sum_{j\in\mathcal{L}} \sum_{r=1}^{R} \hat{v}_{ijt,t+1}(r)\ \tilde{z}_{ijt,t+1}(r). \qquad (41)
\end{aligned}$$

We now assume that the solution in (38)-(40) is not an optimal solution to problem (26). Let $\{x^*_{ijkt} : j \in \mathcal{L}, k \in \mathcal{K}\}$ and $\{z^*_{ijt,t+1}(r) : j \in \mathcal{L}, r = 1, \ldots, R\}$ be an optimal solution to problem (26) giving a strictly higher objective than the solution in (38)-(40). We show that this yields a contradiction. We construct a solution to problem (27) by setting

$$\tilde{y}_{ijt}(r) = \begin{cases} 1 & \text{for } r \le x^*_{ij0t} + x^*_{ij1t} \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all } j \in \mathcal{L},\ r = 1, \ldots, R.$$

It is easy to show that the solution $\{\tilde{y}_{ijt}(r) : j \in \mathcal{L},\ r = 1, \ldots, R\}$ is feasible to problem (27). Since we have $\hat{c}_{ijt}(1) \ge \hat{c}_{ijt}(2) \ge \ldots \ge \hat{c}_{ijt}(R)$, the expression $\sum_{r=1}^{R} \hat{c}_{ijt}(r)\ \tilde{y}_{ijt}(r)$ is equal to the sum of the largest $x^*_{ij0t} + x^*_{ij1t}$ elements of the vector $\{\hat{c}_{ijt}(r) : r = 1, \ldots, R\}$. On the other hand, the expression $\sum_{k=0}^{K} c_{ijkt}\ x^*_{ijkt}$ is equal to the sum of "arbitrary" $x^*_{ij0t} + x^*_{ij1t}$ elements of the vector $\{\hat{c}_{ijt}(r) : r = 1, \ldots, R\}$. Therefore, we have

$$\sum_{r=1}^{R} \hat{c}_{ijt}(r)\ \tilde{y}_{ijt}(r) \ge \sum_{k=0}^{1} c_{ijkt}\ x^*_{ijkt} \qquad \text{for all } j \in \mathcal{L}. \qquad (42)$$

Similarly, since $\hat{v}_{ijt,t+1}(1) \ge \hat{v}_{ijt,t+1}(2) \ge \ldots \ge \hat{v}_{ijt,t+1}(R)$, the expression $\sum_{r=1}^{R} \hat{v}_{ijt,t+1}(r)\ \tilde{y}_{ijt}(r)$ is equal to the sum of the largest $x^*_{ij0t}+x^*_{ij1t}$ elements of the vector $\{\hat{v}_{ijt,t+1}(r) : r = 1, \ldots, R\}$. On the other hand, since we have $\sum_{r=1}^{R} z^*_{ijt,t+1}(r) = \sum_{k=0}^{1} x^*_{ijkt}$, the expression $\sum_{r=1}^{R} \hat{v}_{ijt,t+1}(r)\ z^*_{ijt,t+1}(r)$ is equal to the sum of "arbitrary" $x^*_{ij0t} + x^*_{ij1t}$ elements of the vector $\{\hat{v}_{ijt,t+1}(r) : r = 1, \ldots, R\}$. Therefore, we also have

$$\sum_{r=1}^{R} \hat{v}_{ijt,t+1}(r)\ \tilde{y}_{ijt}(r) \ge \sum_{r=1}^{R} \hat{v}_{ijt,t+1}(r)\ z^*_{ijt,t+1}(r) \qquad \text{for all } j \in \mathcal{L}. \qquad (43)$$

Then, (42) and (43) imply that

$$\begin{aligned}\sum_{j\in\mathcal{L}}\sum_{r=1}^{R}[\hat{c}_{ijt}(r)+\hat{v}_{ijt,t+1}(r)]\ \tilde{y}_{ijt}(r) &\geq \sum_{j\in\mathcal{L}}\sum_{k=0}^{1}c_{ijkt}\ x^*_{ijkt}+\sum_{j\in\mathcal{L}}\sum_{r=1}^{R}\hat{v}_{ijt,t+1}(r)\ z^*_{ijt,t+1}(r)\\ &> \sum_{j\in\mathcal{L}}\sum_{k=0}^{1}c_{ijkt}\ \tilde{x}_{ijkt}+\sum_{j\in\mathcal{L}}\sum_{r=1}^{R}\hat{v}_{ijt,t+1}(r)\ \tilde{z}_{ijt,t+1}(r)=\sum_{j\in\mathcal{L}}\sum_{r=1}^{R}[\hat{c}_{ijt}(r)+\hat{v}_{ijt,t+1}(r)]\ y^*_{ijt}(r),\end{aligned}$$

where the second inequality follows by the assumption that the solution in (38)-(40) is not an optimal solution to problem (26) and the equality follows by (41). The chain of inequalities above implies that $\{y^*_{ijt}(r) : j \in \mathcal{L}, r = 1, \dots, R\}$ is not an optimal solution to problem (27), which is a contradiction. Therefore, the solution constructed in (38)-(40) is an optimal solution to problem (26). This shows that one can construct an optimal solution to problem (26) by using an optimal solution to problem (27). □

References

Albayrak, S. and Krallmann, H. (1995), Artificial intelligence in industrial decision making, control and automation, *in* S. G. Tzafestas and H. B. Verbruggen, eds, '*Distributed Artificial Intelligence in Manufacturing Control*', Kluwer Academic Publishers, The Netherlands, pp. 247–294.

Bellman, R. (1957), *Dynamic Programming*, Princeton University Press, Princeton.

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., Gendreau, M. and Dejax, P. (1993), 'Dynamic and stochastic models for the allocation of empty containers', *Operations Research* **41**, 102–126.

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.

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

Holmberg, K., Joborn, M. and Lundgren, J. T. (1998), 'Improved empty freight car distribution', *Transportation Science* **32**, 163–173.

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

Kurose, J. F. and Simha, R. (1989), 'A microeconomic approach to optimal resource allocation in distributed computer systems', *IEEE Transactions on Computers* **38**(5), 705–717.

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. (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. and Powell, W. B. (to appear), 'Dynamic programming approximations for stochastic, time-staged integer multicommodity flow problems', *INFORMS Journal on Computing* .

Waldspurger, C. A., Hogg, T., Huberman, B. A., Kephart, J. O. and Stornetta, W. S. (1992), 'Spawn: A distributed computational economy', *IEEE Transactions on Software Engineering* **18**(2), 103–117.

Wets, R. J. B. (1974), 'Stochastic programs with fixed recourse: The equivalent deterministic problem', *SIAM Review* **16**, 309–339.