A Dynamic Rail Car Distribution Model with Multiple, Lagged Information Processes

Belgacem Bouzaïene-Ayari* Warren Powell* Huseyin Topaloglu†

*Department of Operations Research and Financial Engineering, Princeton University
Princeton, N.J. 08544
USA.
{belgacem,powel}@princeton.edu

†Department of Operations Research and Industrial Engineering, Cornell University
Ithaca, N.Y. 14850
USA.
huseyin@orie.cornell.edu

Abstract

In this study we present a model and an algorithm for solving the car distribution problem in railway networks. The problem consists of assigning empty cars to customers that are located at different points of the network. The operation is usually planned over a period of several weeks. The problem has to be solved in the presence of multiple information processes with lags between when information becomes known (the phone call) and when it becomes actionable (the car or order is available to be moved). These information processes include both the quantity of cars and orders, but also their attributes (the suitability of the car for a shipper, the destination of an order), and the times required to complete an action. We report computational results of a real case study where the model has been used to solve the problem in a major North American rail road company.

Introduction

The management of empty cars represents one of the basic operations in rail transportation. The main objective consists of covering as many customer demands as possible with empty cars while trying to minimize empty transportation costs and penalties for late service of an order. Railroads must solve this problem in the presence of considerable uncertainty. Most models formulate the problem by either ignoring uncertain events (such as forecasted cars or orders), or treating them deterministically. Several authors have proposed stochastic models which primarily focus on uncertainty in the future customer demands.

This paper is based on an actual project modeling empty car distribution for a major railroad. During this project, it became clear that in addition to uncertainty, there are multiple information streams as well as lagged information processes. Examples of the multiple information streams include:

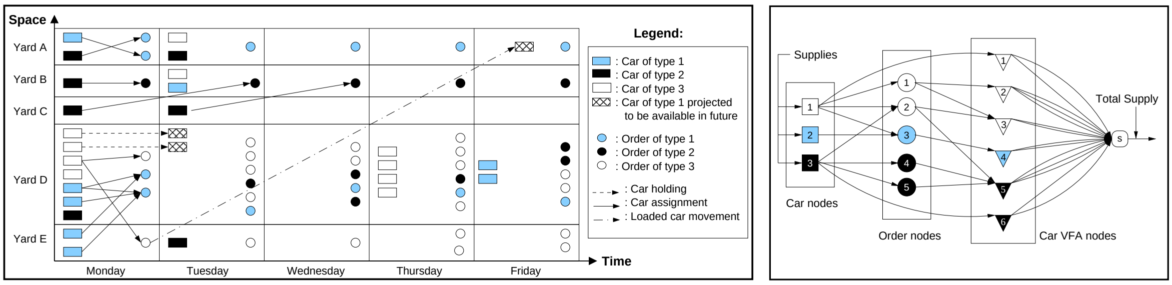
- Car supplies - Cars become available from other railroads.
- Car acceptability - A railroad may move a car to a customer, only to learn that the customer has rejected the car as being unacceptable (dirty, maintenance problems).
- Customer orders - Customers typically make their orders known the previous week, although typically late in the week. The car order consists only of the number of cars required and the preferred type.
- Load times - The time required to load a car is not known until the car is loaded and released back to the railroad.
- Order destinations - The destination of an order does not become known until the car is loaded.
- Transit times - The time required to complete a trip is not known until after the trip is finished.

In addition, there are a number of *lagged* information processes, which arise when there is a difference between when the information is available and when the action actually occurs. The most obvious instance of a lag is the difference between when a customer places an order for a car and when the order has to be served. Other examples are departures of cars - we know at the time of the departure when the car will be available. This second example also highlights that lags can be random. A third example is empty car supplies from other railroads. A railroad can "see" that an empty car on one railroad is headed inbound and will show up on the home railroad at some point in the future.

In addition to the multiple, lagged information processes, the car distribution problem has to respect a variety of business rules:

- The demands of each customer need to be covered from a specified preference list of car types. Some substitution is allowed, but it depends on each individual customer.
- Empty cars can only be stored at primary storage yards.
- Empty car movements are allowed only between certain yards.
- When a car first becomes empty at a customer it *must* be moved to an allowable storage facility.
- An order that should be served on Tuesday may be served later the same week, but the order may not be held over to the subsequent week.

The focus of our project was to optimize the flows of cars over a three week horizon to support tactical planning activities. Figure 1(a) depicts the time-space representation of the problem. The planners want the results of the model to forecast shortages (orders that the railroad will



(a) Time-space representation of the problem.

(b) Graph representation of q_t^n.

Figure 1: Graph representations of the problem.

not be able to cover), perform "what-if" analyses to determine the value of additional cars or the number of cars that can be stored, and plan the flows of empties.

In the following sections we briefly describe the existing solution approaches and the methodology as well as the algorithmic approach that we use in this study to solve the car distribution problem. Then, we present some computational results for a real case where the model is applied to solve the problem of one of the North American railroads. We finally conclude with some possible extensions of the model and future research avenues.

1 Solution approaches and description of the model

We propose a solution approach using approximate dynamic programming where the value of cars in the future are approximated using separable, piecewise linear functions. The strategy builds on prior work applying this strategy for fleet management (see, for example, Godfrey Powell 2002, and Powell and Topaloglu 2003). This prior work considered randomness only in the number of orders being placed, not their attributes. These are critical assumptions, as they allowed the construction of nonlinear value functions that depended on the destination of an order and the time of arrival. In our model, this information is not available.

The solution approach consists of an iterative process that can be described as follows. We discretize the simulation horizon into T equal time periods $[0, \ldots, T]$. At each iteration n, we solve a sequence of subproblems $\{q_t^n\}_{t=0}^T$ where the subproblem q_t^n of time step t is similar to the one illustrated by Figure 1(b). To describe the way the subproblems $\{q_t^n\}$ are constructed and the dynamics of resource and information flows within and between these subproblems, we need to introduce the following notation.

- $\mathcal{C}^R$: Set of resource classes. We have two classes of resources: *car* and *oder*.
- a: Vector of attributes which describe the state of a resource. In our problem, each car has three basic attributes: type, current location and the actionable (available) time. On the other hand, each order has four basic attributes: origin, destination, the actionable (available) time and the list of preferred car types that can be used to cover the order.

$\mathcal{A}^c$: Space of possible outcomes of different vectors of attributes in class $c \in \mathcal{C}^R$.

In the car distribution problem, each order can be satisfied partially and at different times. This allows us to split each order into multiple identical orders where each requires only one car. We use the following notation for the number of resources at different time steps of an iteration n:

$\hat{R}^{c,n}_{tt'a}$: The number of exogenous resources (just entered the system) at iteration n in class $c \in \mathcal{C}^R$ with attribute a that we know about at time t and are actionable at time $t' \geq t$.

$\hat{R}^{c,n}_{tt'}$: $(R^{c,n}_{t,t'a})_{a \in \mathcal{A}^c}$.

$\hat{R}^n_t$: $(\hat{R}^{c,n}_{tt'})_{t' \geq t, c \in \mathcal{C}^R}$.

$R^{c,n}_{tt'a}$: The number of endogenous resources (already in the system) in class $c \in \mathcal{C}^R$ with attribute a that we know about at time t and are actionable at time $t' \geq t$.

$R^{c,n}_{tt'}$: $(R^{c,n}_{ta})_{a \in \mathcal{A}^c}$.

R^n_t: $(R^{c,n}_{tt'})_{t' \geq t, c \in \mathcal{C}^R}$.

The number of resources $R^{c,n}_{tt'a}$ are obtained from three different sources: the resources that are actionable in time $t' \geq t$ and we already know about them at time $t'' < t$, the resources that are actionable in time $t' \geq t$ and we just know about them at time t, and the new exogenous resources that just became knowable at time t and are actionable in time $t' \geq t$. The exogenous resources can also be classified into deterministic and stochastic. To formally write the flow interactions, we define:

ω^n: The realization of all information arriving all time periods at iteration n, i.e., $\omega^n = (\omega^n_1, \ldots, \omega^n_T)$, where ω^n_t is the information arriving during $[t, t+1)$.

Ω: Set of all possible sample information realizations ($\Omega = \{\omega\}$).

d: A type of decision that acts on a resource or a class of resources. In our problem, d can be *assign* a car to an order, *hold* a car or an order to the next time step, *move empty* a car to a storage facility.

$\mathcal{D}$: The set of possible decisions just enumerated.

x^n_{tad}: The quantity of resources with attribute a acted on with a decision $d \in \mathcal{D}$ at iteration n at time t.

x^n_t: $(x^n_{tad})_{a \in \mathcal{A}, d \in \mathcal{D}}$.

x^n: $\{x^n_t\}_{t \in [0,\ldots,T]}$.

$M_t(a, d, \omega^n)$: The *transition* function that models the outcome of a decision d acting on a resource with an attribute vector a at time t requiring time τ to complete and produce a resource with an attribute vector a' at time $t+\tau$, i.e., $M_t(a, d, \omega^n) = (a', \tau)$. The information ω^n would contain, for example, the information about the order destination after assigning a car to it.

It is important to note that the assign decision includes at most five sub-decisions: *move empty* to the order location if the car is not at the order location, *hold* the car or the order if the car arrives before or after the order actionable time, *load* the car, *move loaded* car to the order destination, and *unload* the car at the destination. The actionable time and the location of the car at the end of an assign decision are stochastic and are unknown when the decision is made since both the travel times and the order destination are not known in advance. Figure 2 illustrates the arrival of new information needed to implement an assign decision over time.

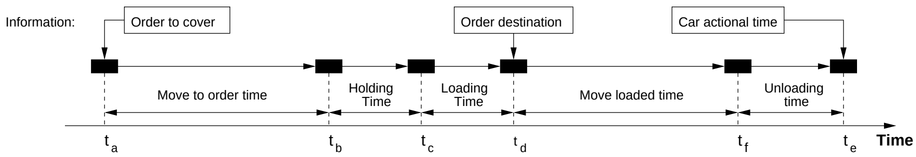


Figure 2: Information arrivals during the implementation of an assign decision.

In the iterative process, the resource quantities that we know about at time step t and are actionable at $t' \geq t$ are viewed as a function of the information $(\omega_0^n, \ldots, \omega_{t-1}^n)$ collected so far before time t. At time step 0, all prior information is captured by the state variable R_0.

After solving a subproblem q_t^n, the resource quantities that we know about at $t+1$ that will be actionable at time $t' \geq t+1$ are computed as follows:

$$R_{t+1,t'a'}^n(\omega^n) = R_{tt'a}^n(\omega^n) + \hat{R}_{t+1,t'a}^n(\omega^n) + \sum_{a \in \mathcal{A}} \sum_{d \in \mathcal{D}} \delta_{t'a'}(a, d, \omega^n) x_{tad}^n, \ \forall\ t' \geq t+1, \tag{1}$$

where $\delta_{t'a'}(a, d, \omega^n)$ is an indicator function defined as follows:

$$\delta_{t'a'}(a, d, \omega^n) = \begin{cases} 1, & \text{if } M(a, d, \omega^n) = (a', t' - t), \\ 0, & \text{Otherwise.} \end{cases} \tag{2}$$

Note that the last term of Equation (1) computes the quantity of resources that we acted on at time step t after the implementation of x_t^n that will actionable at a future time step $t' \geq t+1$. This information becomes knowable after the implementation of the solution x_t^n.

The supplies of subproblem q_t^n are cars that we know about at time t that will be actionable at time $t' \geq t$ and are given by $R_{tt'a}^{car,n}(\omega^n)$. Similarly, the demands of q_t^n are the orders that we know about at time t that will be actionable at time $t' \geq t$ and are given by $R_{tt'a}^{order,n}(\omega^n)$. For computational reasons, we can include in q_t^n only the resources that are actionable in the near future by limiting the time t' to be the interval $[t, \ldots, \min(t + T^{ph}, T)]$, where T^{ph} is a time parameter that we name the *planning horizon.*

When the outcome of a decision is deterministic, an effective approximation is to solve subproblems of the form:

$$\max_{x_t} \sum_{a \in \mathcal{A}} \sum_{d \in \mathcal{D}} c_{tad} x_{tad} + \sum_{a' \in \mathcal{A}} \bar{V}_{ta'}^{n-1}(R_{t+1,a'}(x_t)) \tag{3}$$

where:

$$R_{t+1,a'}(x_t) = \sum_{a \in \mathcal{A}} \sum_{d \in \mathcal{D}} \delta_{a'} x_{tad} \tag{4}$$

Equation (3) uses a separable value function approximation (VFA) which we represent as $\bar{V}_{ta'}^{n-1}(R_{t+1}(x_t))$. In our research, we show how to modify this approximation to handle uncertainty in the modify function. Instead of a function indexed by the outcome a', we use a value function indexed by the attributes that we know when we make a decision. For example, when we assign a car to move an order but do not know the destination of the order, we index the value function by the origin of the order.

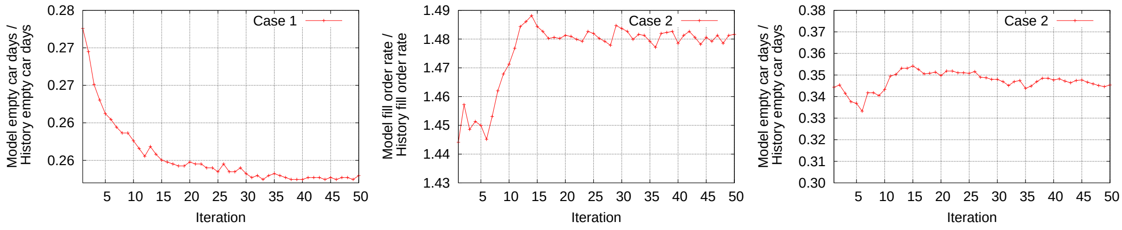
Algorithm 1 summarizes the solution approach described in this section for the car distribution problem. In Algorithm 1, $\hat{V}_t^n$ denotes the set of newly constructed car VFAs when solving q_t^n, $\bar{V}$ denotes the set of all VFAs, and I_t^n denotes the information collected after solving q_t^n that is required to update the elements of $\bar{V}$.

Algorithm 1 Solution algorithm

$n \leftarrow 1$, $\bar{V} \leftarrow \emptyset$, stop $\leftarrow$ false
while stop is false **do**
  $x^n \leftarrow \emptyset$, $I^n \leftarrow \emptyset$
  Sample ω^n {here, we sample the quantities of stochastic exogenous resources for the all time periods and we add them with the deterministic resources to the system. We also sample the destination of each order. This information will be used only after a car assignment is made and the car is ready to move to the order destination}
  $R_{0t'}^n \leftarrow \hat{R}_{0t'}^n$
  for $t \leftarrow 0$ to T **do**
    Construct subproblem q_t^n where supplies and demands are obtained by $R_{tt'}^n(\omega^n)$
    Solve q_t^n, i.e., find x_t^n
    Implement x_t^n {here, we compute $R_{t+1,t'}^n(\omega^n)$ using Equation (1)}
    Collect $\hat{V}_t^n$ and I_t^n {I_t^n are usually the gradients collected after solving q_t^n}
    $x^n \leftarrow x^n \cup x_t^n$, $I^n \leftarrow I^n \cup I_t^n$, $\bar{V} \leftarrow \bar{V} \cup \hat{V}_t^n$
  end for
  Update each element of $\bar{V}$ using I^n {smooth the VFAs using a step size $\alpha \in (0, 1]$}
  if the stopping criterion is met **then**
    stop $\leftarrow$ true, the "optimal" solution is x^n
  else
    $n \leftarrow n + 1$
  end if
end while

2 Computational results

The algorithm described in the previous section has been implemented in Java and used in a *deterministic* mode (car supplies, order demands and order destinations are known in advance) to solve the car distribution problem of a major railroad. The company has a fleet of about 1800 cars of 11 types spread over 208 yards. The car assignment operation is done over a three week simulation horizon where a total demand of 4600 empty cars of different types need to be satisfied. The decision horizon has been set to one day and the planning horizon to four days. Two datasets of the same simulation period have been considered. The first used orders from history, and the second used forecasted orders.



(a) Empty car days, Historical orders.

(b) Fill order rate, Forecasted orders.

(c) Emtpy car days, Forecasted orders.

Figure 3: Model results compared to history.

The historical run produced a fill order rate (the percent of demands that were covered) of 100 percent in all iterations. The empty car days per moved order from the model using the historical dataset divided by the empty car days from history over the iterations are given in Figure 3(a). Notice the steady improvement of the empty car days over the first 40 iterations to reach a steady level which is low compared to history. In the run on forecasted data, the model outperformed history for both the fill order rate and the empty car days (see Figures 3(b)-3(c)). However, the improvement is more noticeable for the fill order rate where the model moves about 48 percent more cars than in history.

In our presentation, we will present results of experiments where both the number of orders and their attributes are random. We believe this is the first formulation of a stochastic fleet management problem which exhibits randomness not only in the quantities of orders, but also their attributes.

References

G. Godfrey and W.B. Powell, "An Adaptive, Dynamic Programming Algorithm for Stochastic Resource Allocation Problems I: Single Period Travel Times", *Transportation Science* **36**(1), 21-39 (2002).

W.B. Powell and H. Topaloglu, "Stochastic Programming in Transportation and Logistics", in *Handbook in Operations Research and Management Science*, Volume on Stochastic Programming, A. Ruszczynski and A. Shapiro (eds), 555-635, Elsevier, Amsterdam, 2003.