Real-Time Optimization of Containers and Flatcars for Intermodal Operations

Warren B. Powell
Tassio A. Carvalho

Department of Civil Engineering
and Operations Research
Princeton University
Princeton, NJ 08544

Statistics and Operations Research
Technical Report SOR-96-05

February 6, 1998

Abstract

We propose a dynamic model for optimizing the flows of flatcars that considers explicitly the broad range of complex constraints that govern the assignment of trailers and containers to a flatcar. The problem is formulated as a *logistics queueing network* which can handle a wide range of equipment types and complex operating rules. The complexity of the problem prevents a practical implementation of a global network optimization model. Instead, we formulate a global model with the specific goal of providing network information to local decision makers, regardless of whether they are using optimization models at the yard level. Thus, our approach should be relatively easy to implement given current rail operations. Initial experiments suggest that a flatcar fleet that is managed locally, without the benefit of our network information, can achieve the same demand coverage as a fleet that is 10 percent smaller, but which is managed locally with our network information.

We consider the problem of managing a fleet of railroad flatcars over a network in a real-time setting. A local terminal manager must determine how to assign a set of trailers and containers to the available flatcars that can move this equipment. The challenge is that there are several dozen types of flatcars, and many types of trailers and containers. Each flatcar is best designed for moving a certain configuration of trailers and containers. For example, some flatcars do not have the proper connections to hold a container. Other flatcars might hold four 45 foot containers, but only two refrigerated 45 foot containers. Other flatcars can only hold two containers, and it does not matter if they are refrigerated or not. The yard manager needs to find groups of trailers and containers with the same destination that best utilize the *hitch capacity* of each flatcar.

The problem of assigning trailers and containers (or, more simply, *boxes*) to flatcars at a yard to maximize hitch utilization is itself a challenging problem. It can be solved greedily or using an optimization procedure (see, for example, Feo & Gonzaliez-Velarde (1995)). However, both approaches are myopic and ignore the ability of the destination to properly utilize the equipment. For example, we may only have containers at a yard, but we have a flatcar that can hold a trailer (a special hitch is needed to accommodate trailers). If the container is going to a destination that originates a lot of trailers, then we want to favor flatcars that can carry trailers.

In this paper, we focus on methods that contribute network information into local yard decisions. It is often the case that different flatcars can be used to move a set of trailers and containers to a destination which, from the perspective of the local yard operation, appear equally efficient. For example, a yard manager might use a flatcar that can move either trailers or containers to move two containers to their destination. That destination, however, may have little use for flatcars that move trailers, whereas another yard desperately needs this equipment type. It is the goal of this paper to introduce network level information to improve the decisions made at a local level so that they better optimize an entire network.

Railroads were the first transportation industry to use optimization models to describe their networks and the flow of empty cars. However, these models were quite simple. The first work that uses a space–time network is by White & Bomberault (1969) (see also Shan (1985), and the survey by Assad (1987)). Chih (1986) develops a multicommodity network flow model for railcar distribution that uses decomposition techniques to reduce the flows of multiple car types to single commodity network flows. Joborn (1995) presents a multicommodity network flow model that considers capacity restrictions on trains for repositioning of empty freight cars.

Not surprisingly, there appears to be very little optimization literature focused on intermodal transportation. Crainic, Gendreau & Dejax (1993) present a sequence of dynamic models (single commodity deterministic, multicommodity deterministic, single commodity stochastic) for empty allocation of containers. These models determine the number of empty containers that must be relocated among several locations. However, they do not deal with determining how containers will be repositioned, and if by rail, how to assign them to flatcars. The only work that we are familiar with that deals explicitly with the problem of assigning trailers and containers to flatcars is Feo & Gonzaliez-Velarde (1995), which considers only the local problem at a single yard, at a single point in time.

For our work, we take advantage of a new formulation of the dynamic fleet management problems developed by Powell & Carvalho (1997*a*) called the logistics queueing network (LQN). This approach is extended to multicommodity problems in Powell & Carvalho (1997*b*). An important feature of this modeling approach is the flexibility it provides in modeling complex operations by decomposing large dynamic problems into sequences of very small problems that deal with one location at a time, one time period at a time. This paper tests this feature by showing how it can be used to plan what is perhaps the most complex fleet management problem that arises in any practical application: the management of a fleet of flatcars for a railroad.

This paper makes the following research contributions: 1) We introduce the first computationally tractable model for optimizing the flows of flatcars over space and time which explicitly handles the complex constraints governing the assignment of trailers and containers onto flatcars. Features of the system include integer solutions, real–time implementation and explicit handling of both central and local decision making. 2) We show that the new model can reduce costs by three to five percent over myopic decisions that focus purely on maximizing flatcar utilization at each yard. 3) We present an integrated framework for simultaneously optimizing the flows of a railroad's own trailers, the trailers and containers of customers and the movement of flatcars.

This paper represents both an extension of the methodology in Powell & Carvalho (1997*a*) and Powell & Carvalho (1997*b*) and an application to intermodal operations. Reference Powell & Carvalho (1997*a*) considers a single commodity problem where a single resource (such as a trailer) serves a single task (such as a load of freight), and where the resources are homogeneous (for example, all the trailers are the same). Reference Powell & Carvalho (1997*b*) considers the multicommodity case, where the resources are heterogeneous, representing, for example, different

types of boxcars. However, it is still assumed that one resource serves one task. In this paper, we consider the problem where one resource (a flatcar) may serve multiple tasks (several trailers and containers) with complex rules governing what combinations of trailers and containers can be assigned to a particular flatcar.

The paper is organized in eight sections. In the first section we provide background material into the intermodal flatcar management problem. In section 2 we propose a modular framework that results in two interconnected dynamic resource allocation models, one to optimize the flows of trailers and containers owned by the railroad, and the second to optimize the flows of the flatcars. Optimizing the flows of trailers and containers is a direct application of the LQN methodology given in Powell & Carvalho (1997*b*), which is reviewed in section 3. The application of this method to optimizing the flows of railroad-owned trailers and containers is given in section 4. The second model matches flatcars to sets of boxes and is described in section 5. Section 6 describes the experiments that were performed to tune the algorithm and estimate the value of network information on local decision making. In section 7 we discuss implementation issues. Finally, we present the conclusion in section 8.

1 Problem Description

There are a variety of issues that arise in flatcar management that need to be appreciated. In this section, we provide brief discussions of network operations, flatcar equipment characteristics, trailer and container characteristics, and yard operations. We also provide a summary of the problem of managing boxes owned by the railroad that must also be repositioned, competing for space on available flatcars.

Network operations

The rail network is represented by a set of services representing trains moving between originating, transfer and terminating points in the railroad. A train service may start at terminal A with D as a termination, with intermediate stops at B and C where cars may be added or dropped. A terminal may be interior to the network, where freight may originate or terminate, or be transferred from one train to the next. If the terminal is at a port, we might expect a high volume of container traffic that is transferring with ships. If it is inland but at a common point with another railroad, there might be a high volume of rail interchange traffic, which has the characteristic of

being very difficult to forecast (because of the difficulty railroads have in sharing information). Inland terminals could be expected to have a high percentage of trailers as freight is brought to a yard by truck.

Decision making in a railroad is typically made hierarchically. Some decisions are made centrally, governing primarily the repositioning of empty flatcars. Others are made locally. In particular, the decision of what flatcar to use for a particular set of trailers and containers requires a knowledge of local yard operations that is not available centrally. The solution approach we propose in this paper is especially well adapted to supporting decision making in a decentralized environment.

Yard operations

The process of physically loading boxes on flatcars starts with the yard supervisor forming a train out of a sequence of flatcars. Some of these may already have boxes on them, while the rest will be empty. The yard superviser tries to load trailers and containers to contiguous *blocks* of flatcars with a common destination. This is not always possible, but represents a major priority for yard supervisors. Since flatcars arrive and are stored in segments that are already connected together, the goal of keeping these cars hitched, and sending them to the same destination, is the single most important factor limiting the flexibility of the local yard master.

Flatcar characteristics

An important characteristic of the flatcar management problem is the tremendous variety of flatcars, along with the variety of trailers and containers that the flatcars need to move. Figure 1 illustrates some different loading configurations of boxes on flatcars. Flatcars feature slots where containers may be loaded, with flatcars ranging between one and as many as seven slots (using spine cars). It may be possible to double stack containers within a slot (of course, you cannot double stack trailers). It may also be possible to put two short (28 foot) containers on the bottom, and a single long container on the top, allowing as many as three containers to be put into a single slot. Using two short containers in one slot may not be possible if they are refrigerated, since the refrigeration unit adds to the length of the container. Other rules govern the configuration, such as the restriction that you cannot put an empty container on the bottom when they are double stacked. Finally, the only flatcars that can carry trailers are those that have special hitches to keep the trailers from rolling.

Railroads have tended in the past to purchase flatcars that are productive for certain markets,

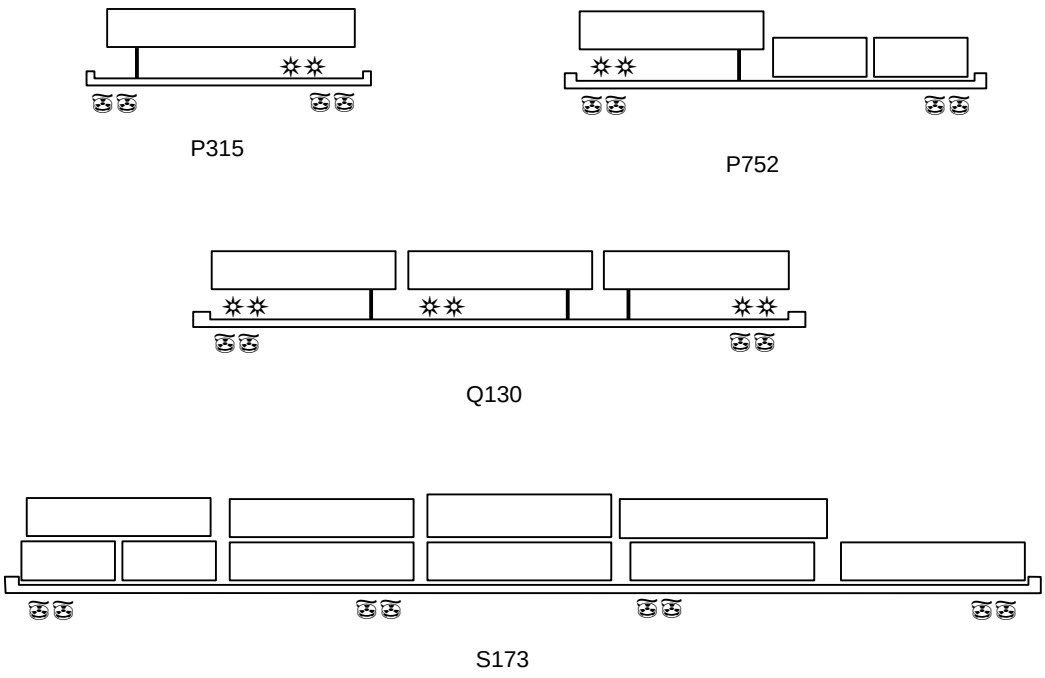

Figure 1: Several types of flatcars loaded with trailers and containers.

creating a tremendous mix of flatcar types. In one dataset, we counted over 100 distinct types of flatcars, with over 40 appearing regularly.

A different dimension to modeling flatcars is their availability. At any point in time, flatcars may be sitting available at a yard, enroute empty to a different location, enroute loaded to a destination, or it may be in the possession of another railroad (with, we assume, an unknown status). A container that is enroute empty from, say, A to D can, in principle, be rerouted at an intermediate terminal, whereas a loaded container must finish its task (boxes are not unloaded enroute and reloaded onto a different flatcar). Loaded flatcars may be destined to a location on a railroad's own network, or it may be headed off-line. In the latter case, we assume the flatcar is "lost" to the system. Finally, flatcars that are off-line, in the control of another railroad, may suddenly "reappear" at an interchange point, either empty (at which point they are immediately available) or loaded (in which case they will become available at some point interior to the railroad at a later time). One of the challenges of managing flatcars is estimating the availability of flatcars entering the system from off-line.

Trailers and containers

There is also a wide variety of trailers and containers. Containers have two standard widths and heights and several standard lengths, varying from 20 to 57 feet. Trailers also have two standard widths and heights, and lengths varying from 26 to 53 feet, and may be refrigerated. The real world adds to this diversity; after being used for several years, containers might be involved in small accidents and undergo repairs, losing a few inches or feet in the process.

The demand for flatcars arises in the form of requests to move boxes (loaded or empty) between pairs of terminals. Requests to move boxes may be immediate (move the box the same day it is requested) or may be made a week or more in advance. Frequently, the requests to move boxes immediately are often high priority, from your most important customers. When we make decisions about moving flatcars loaded or empty, we have to look a week or more into the future in order to incorporate downstream effects. As a result, we must combine known customer requests with forecasted demands.

In our work, we use a special forecasting system developed specifically for estimating daily freight demand (see Godfrey & Powell (1995)). The system incorporates multiple calendar effects and estimates booking profiles that gives the number of days into the future that a customer will call in a request. This allows us, then, to combine both known and forecasted requests to move boxes, over horizons that extend a week or more into the future.

Railroad-owned equipment (ROE)

The problem of deciding how to move flatcars is typically coupled with the problem of planning the movements of boxes owned by the same railroad. We call these boxes *railroad-owned equipment* (ROE), which can be leased to customers. The steps that railroad–owned boxes go through in the process of moving loaded is shown in figure 2. After terminating in an empty pool, the trailer may wait for reassignment, or be repositioned empty to another pool. At any given rail terminal, there must be a stock of empty trailers to satisfy customer requests. Once a box is assigned to a request, this box is removed from the stock to be loaded by the customer. It returns to the railroad a few days later to transport the freight to its destination. Most often, a box is then returned to the pool of empty boxes. A box can also be loaded at the same location where it was unloaded and bypass the stock of empty boxes. The demand for empty boxes is not evenly distributed across the network. The railroad must manage the supply of empty boxes at each terminal, relocating them when necessary. Thus, these empty boxes compete for space on flatcars with other trailers and containers.

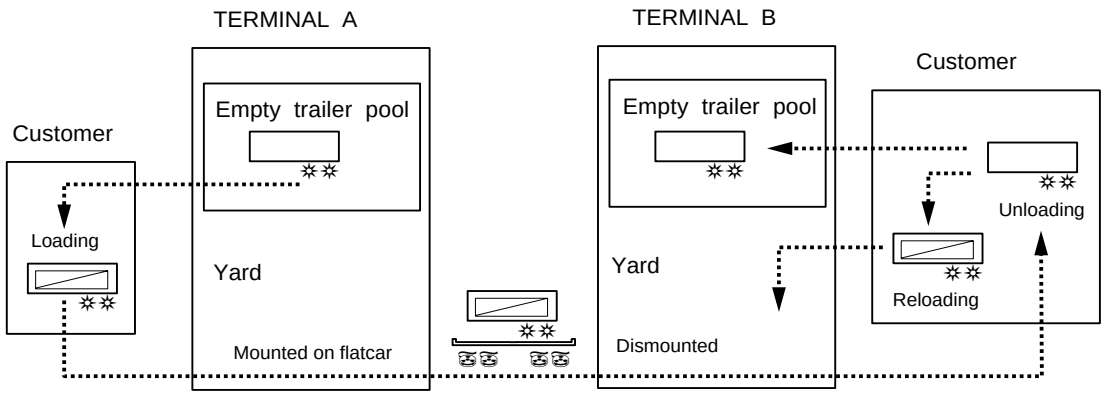

Figure 2: Steps in a loaded movement for railroad–owned trailers.

In theory, the allocation of empty ROE equipment, and the management of empty flatcars, is a problem that should be solved jointly. While this is possible, it was not our feeling that it could be successfully implemented at this point in time. Instead, we solved the problem sequentially. We first optimize the movements of ROE equipment, and then add the planned movement of empty equipment to the customer-driven demands to move loads.

The management of railroad–owned equipment is a classical dynamic fleet management problem. The decision variables for the ROE model are the number of empty trailers of each type that must be moved between each terminal pair at each time period. The objective function is to maximize the return that results from satisfying requests, given by a revenue (or reward) from satisfying a request minus variable operating costs. The problem has been widely studied over the last 40 years, with models and algorithms proposed by Feeney (1957), White (1972), Herren (1973),Herren (1977), Turnquist (1986), Mendiratta (1981), Chih (1986), Shan (1985), and Powell (1988). A review of a broad range of dynamic models for fleet management is given in Powell, Jaillet & Odoni (1995).

The ROE model is most often formulated as a static, myopic model which ignores all future events. In practice, there are two forms of uncertain information that can and should be incorporated. The first is customer demands that are not known at a particular point in time. The second is the flow of empty trailers and containers as they enter the system either from customers or from other railroads. We incorporate both types of future events.

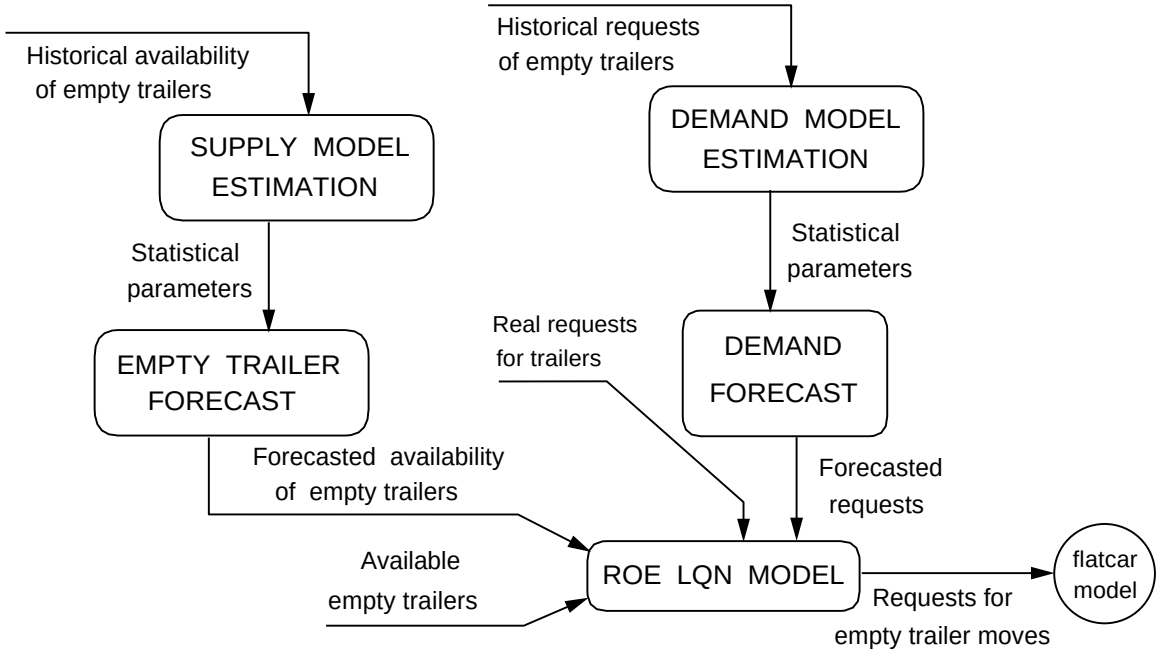

Figure 3: Modules for the railroad–owned equipment model.

2 Modeling Approach

As explained in section 1, to keep the complexity of the overall problem at an acceptable level, we work with two dynamic assignment models: one assigns trailers to customer requests and the other assigns flatcars to boxes. While the ROE model is similar to the multicommodity LQN, a special LQN model is presented for the flatcar assignment in order to deal with the consolidation of trailers and containers on flatcars.

In the context of resource allocation, the requests for trailers can be viewed as tasks that must be satisfied by resources, which are the empty trailers themselves. Even though trailers have different widths and heights, requests for empty trailers only specify length. We have chosen to use one multicommodity LQN model in order to allow for size substitutions.

Figure 3 shows the modules that compose the ROE model. We use historical patterns (of both customer requests and the exogenous arrivals of empty trailers) to fit a statistical forecasting model. We then use the forecasting model to generate a forecast of future events (either new demands entering the system, or the availability of new capacity). In our work, we generate only a single forecast (equivalent to a deterministic model) but our methodology does not "abuse" our

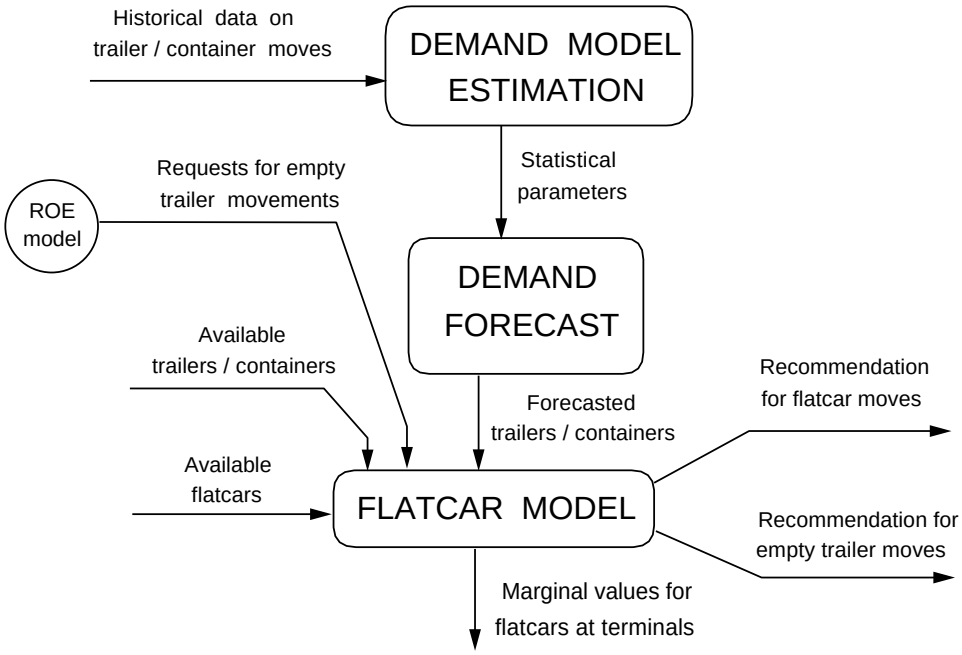

Figure 4: Modules for the flatcar model.

ability to see the future. We could easily do repeated samples and obtain an explicit stochastic model.

The output of the ROE model consists of a series of recommended movements for the empty trailers. These movements satisfy requests that are on hold and replenish the stock of containers at some locations based on the expectation that requests will appear and will be fed to the flatcar model.

In the flatcar model, the movement of a box is regarded as a task and the flatcar is the resource to accomplish it. As in the ROE model, we look at the problem of assigning tasks to resources over a planning horizon. As the information on boxes available within the whole horizon is not available, we must rely on historical data to generate a random sample of boxes that will become available within the planning horizon. Thus, the flatcar model works with three types of demand (see figure 4). The first type is the set of real demands, i.e., loaded boxes waiting to be moved, either belonging to the railroad or to a third party. The second type of demand is composed of the requests for moving empty trailers belonging to the railroad that come out of the ROE model. The third set of demands is the random realization of demands that comes out of the forecast module. The flatcar model does not distinguish whether a given box is real or forecasted.

The resource set for the flatcar model is composed of the available flatcars and those that may become available in the future. The flatcar model provides three kinds of output. First, it outputs

suggested flatcar flows, by type, terminal pair and departure time. Second, it outputs suggested flow of empty trailers, again by type, terminal pair and departure time. This kind of output is somewhat like the output of the ROE model, but now validated by the "competition" for space on the flatcars. Third, it outputs marginal values of flatcars by type, terminal and time. The module containing the flatcar model is described in section 5.

3 Summary of the LQN Solution Approach

We have introduced a new approach, the Logistics Queueing Network (LQN) approach, to solve dynamic resource allocation problems in Powell & Carvalho (1997*a*). This approach is extended to consider different types of resources in Powell & Carvalho (1997*b*). It is shown that it delivers integer solutions that are very close to the optimal value at a fraction of CPU time needed by linear solvers. In this section we present a short summary of this approach for multicommodity problems, which we then directly apply to the solution of the railroad–owned equipment problem. We then use this solution approach to devise an method for optimizing the flows of flatcars, which is a much harder problem.

Our modeling approach is to optimize activities over discrete time intervals $t = (0, 1, \ldots, T)$, where T is the length of the planning horizon. Points in space are represented using the indices i and j, and a point in space time is represented by (i, t). The travel time from i to j, represented by τ_{ij}, gives the number of time periods required to travel from i to j. A movement from city i at time t to city j, arriving at time $t + \tau_{ij}$, is represented by the link (ijt). If x is a vector, then x_{ijt} is the $(ijt)^{th}$ element of this vector, and x_t is the subvector consisting of all the elements (ijt) for a single t. Commas are used to separate indices only when an expression is involved, as in $x_{i,j,t+\tau_{ij}}$.

We now introduce the following notation:

- $\mathcal{C}$ is the set of terminals i in the network.
- $\mathcal{A}$ is the set of vehicle types.
- τ_{aij} is the travel time for vehicle type $a \in \mathcal{A}$ between terminal $i \in \mathcal{C}$ and terminal $j \in \mathcal{C}$.
- $\mathcal{N}$ is the set of nodes $(i, t), i \in \mathcal{C}$, $t \leq T$, in the dynamic network.
- $\mathcal{B}$ is the set of load types.

- Υ_{ab} is the indicator variable giving feasible load/vehicle pairs, where $\Upsilon_{ab} = 1$ if vehicle type a can be assigned to load type b.
- $\mathcal{L}$ is the set of loads l of all types available within the planning horizon, T.
- $\mathcal{L}^b$ is the set of loads l of type b, for each type $b \in \mathcal{B}$, available within the planning horizon, T.
- $\mathcal{T}_l$ is the set of feasible departure times for satisfying load $l \in \mathcal{L}$, otherwise known as the *departure time window*.
- $\mathcal{L}^b_{ijt}$ is the set of loads $l \in \mathcal{L}^b$ with origin i and destination j having t as a feasible departure time.
- R_{ait} is the net inflow ($R_{ait} > 0$) or outflow ($R_{ait} < 0$) of vehicles of type a at terminal i at time t.
- r_{alt} is the profit generated by choosing vehicle type a and time t to satisfy load l.
- c_{aij} is the cost of repositioning one empty vehicle type a over link (i, j, t).

The decision variables are

- $x_{alt} = 1$ if load l is served by vehicle type a at time t, and 0 otherwise.
- $z_l = 1$ if load l is never served within the time window, and 0 if the load l is served within the time window. z_l plays the role of a slack variable.
- y_{aijt} is the number of vehicles type a being repositioned empty along link (i, j, t). If $i = j$, y_{aiit} represents the number of vehicles of type a in inventory at terminal i from time t to time $t + 1$.
- w_{aijt} is the total flow of vehicles of type a on the link (i, j, t) .

The problem is formulated as a mixed–integer linear program. The objective function we are maximizing is stated as:

$$G(x, y) = \sum_{t=0}^{T} \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}} \sum_{a \in \mathcal{A}} \left(\sum_{b \in \mathcal{B}} \sum_{l \in \mathcal{L}^b_{ijt}} r_{alt} x_{alt} - c_{aij} y_{aijt} \right) \tag{1}$$

This problem can be formulated as:

$$\max_{x,y} \; G(x,y) \tag{2}$$

subject to:

$$\sum_{t\in\mathcal{T}_l}\sum_{a\in\mathcal{A}} \Upsilon_{ab} x_{alt} + z_l \;=\; 1 \qquad \forall\, l \in \mathcal{L}^b \;\; \forall\, b \in \mathcal{B} \tag{3}$$

$$\sum_{b\in\mathcal{B}}\sum_{l\in\mathcal{L}^b_{ijt}} \Upsilon_{ab} x_{alt} \; + y_{aijt} - w_{aijt} \;=\; 0 \qquad \forall\, i,\; j \in \mathcal{C},\; \forall\; t \;\leq T\;,\; \forall\; a \in \mathcal{A} \tag{4}$$

$$\sum_{j\in\mathcal{C}} w_{aijt} \; - \sum_{j\in\mathcal{C}} w_{ajit-\tau_{aij}} \;=\; R_{ait} \;\; \forall\,(i,t) \in \mathcal{N},\;\; \forall\, a \in \mathcal{A} \tag{5}$$

$$y_{aijt}\,,\; w_{aijt} \;\geq\; 0 \tag{6}$$

$$x_{alt} \;=\; (0,1) \tag{7}$$

where constraints (3) restrict the maximum number of vehicles to be assigned to each load to one and constraints (4) and (5) enforce flow conservation for each vehicle type at each node.

The LQN approach results from: 1) stating the objective function in a recursive form, 2) approximating the future value function at each time stage as a linear function of the number of vehicles available at each node and 3) constraining the unbounded decision variables (in this case, y) by upper bounds. The result is a decomposition of the problem (2) –(7) into *local problems*. To present the solution algorithm, we first define:

- $\bar{\mathcal{L}}_{it}$ is the set of loads l with origin i, which are available to move at time t and have not been moved at a time prior to time t at a given solution.
- $\mathcal{L}^0_{it}$ is the set of loads l with origin i, where t is the beginning of the time window $\mathcal{T}_l$. Thus, $\mathcal{L}^0_{it}$ is the set of loads that first become available to move at time t.
- $\mathcal{L}^f_{it}$ is the set of loads l with origin i, where t is the end of the time window $\mathcal{T}_l$. $\mathcal{L}^f_{it}$ is the set of loads that will expire from the system if they are not moved at time t.
- V_{ait} is the number of vehicles of type a available at node (i,t).
- u_{aijt} is the upper bound on y_{aijt}.
- ξ_{ait} is the spatial potential function for vehicles, i.e., it is a measure of how desirable is to have one more vehicle of type a at a node (i,t).

In the recursive form, the objective function for stage t appears as

$$G_t(V_t\,,\mathcal{L}_t\,) \;=\; \sum_{i\in\mathcal{C}} g_{it}(x\,,y,V_{it}\,,\mathcal{L}_{it}\,) + G_{t+1}(V_{t+1}\,,\mathcal{L}_{t+1}\,) \tag{8}$$

where g_{it} represents the value of the decisions taken at node (i,t). We use an approximation for the value function G_{t+1} such that

$$\hat{G}_t(V_t\,,\mathcal{L}_t\,) \;=\; \sum_{i\in\mathcal{C}} g_{it}(x\,,y,V_{it}\,,\mathcal{L}_{it}\,) + \xi_{t+1}V_{t+1} \tag{9}$$

This results in the decomposition of the problem. The local problem at node (i,t) is represented by:

$$\max_{x_t,y_t} \sum_{j\in\mathcal{C}}\sum_{a\in\mathcal{A}} \left(\sum_{b\in\mathcal{B}}\sum_{l\in\mathcal{L}^b_{ijt}} (r_{alt}+\xi_{a,j,t+\tau_{aij}})x_{alt} - (c_{aij}-\xi_{a,j,t+\tau_{aij}})y_{aijt} \right) \tag{10}$$

subject to:

$$\sum_{a\in\mathcal{A}} \Upsilon_{ab}x_{alt} \;\leq\; 1 \qquad \forall l\in\bar{\mathcal{L}}_{it} \tag{11}$$

$$y_{aijt} \;\leq\; u_{aijt} \quad \forall j\in\mathcal{C},\ \forall a\in\mathcal{A} \tag{12}$$

$$\sum_{l\in\bar{\mathcal{L}}_{it}}\sum_{a\in\mathcal{A}} x_{alt} + \sum_{j\in\mathcal{C}} y_{aijt} \;\leq\; V_{ait} \quad \forall a\in\mathcal{A} \tag{13}$$

$$x_{alt}\,,\ y_{aijt} \;\geq\; 0 \tag{14}$$

where the set of loads, $\bar{\mathcal{L}}_{i,t+1}$, and the number of vehicles, $V_{a,i,t+1}$, available at the next time period are computed by:

$$\bar{\mathcal{L}}_{k,t+1} \;=\; \{\bar{\mathcal{L}_{kt}} \setminus \{\mathcal{L}^s_{kt}\cup\mathcal{L}^f_{kt}\}\}\cup\mathcal{L}^0_{k,t+1} \qquad \forall k\in\mathcal{C} \tag{15}$$

$$V_{a,k,t+1} \;=\; V_{akt} - \sum_j w_{akjt} + \sum_i w_{aikt+1-\tau_{aij}} + R_{ait+1} \quad \forall k\in\mathcal{C},\ \forall a\in\mathcal{A} \tag{16}$$

where the set $\mathcal{L}^s_{k,t}$ is the set of loads with origin i that were assigned a vehicle at time t.

Each local problem is equivalent to a static assignment problem, involving the assignment of a vehicle of type a to a task of type b. In order to improve speed, we do not solve the local problem to optimality but use a greedy heuristic (see Powell & Carvalho (1997*b*)) to find a good feasible

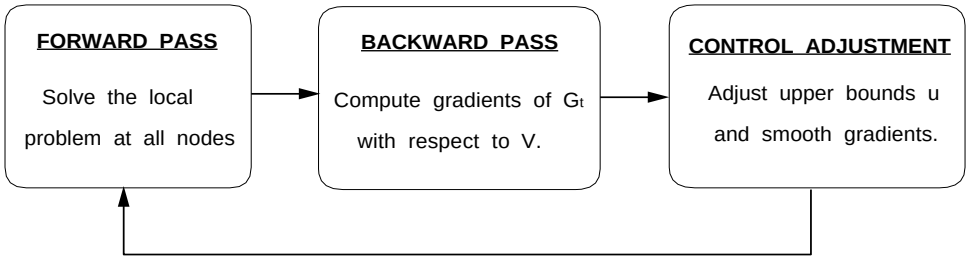

Figure 5: The basic algorithm for the LQN approach.

solution. In this heuristic, we compute all assignment costs c_{ab} and store them in a heap. We then chose the link with the lowest cost and assign a flow of one as long as it is feasible to do so (that is, assuming vehicle a has not yet been assigned and that the load b has not yet been covered). For the flatcar problem, this heuristic proved essential, since we assign a *configuration* of boxes to a flatcar. For this problem, there does not exist an equivalent assignment problem.

Two kinds of variables control the solution of the local problems: the upper bounds on empty moves, u, and the spatial potential function, ξ. These control variables can be adjusted with the help of gradients of the objective function with respect to the supply of vehicles at each node. This results in the iterative algorithm outlined in figure 5. The forward pass consists of solving the local problem at each node. The backward pass consists of finding values for the gradients ν_{it}, defined as

$$\nu_{ait} = \frac{\partial \hat{G}_t}{\partial V_{ait}} \tag{17}$$

A brief description of the forward pass, backward pass and control adjustment follows.

Forward Pass: The forward pass starts by solving the local problem (10)–(14) for each terminal for the first time period. Then equations (15) and (16) are used to find the number of vehicles and the sets of loads available at each node for the second time period. Then the local problems for the second time period can be solved. This procedure continues to the end of the planning horizon.

Backward Pass: This module computes approximations for the gradients ν_{ait} by using finite differences. These gradients capture the incremental value of another vehicle of type a at time

(i,t). We first show how the right gradient is computed. In order to represent the perturbation on the vector of vehicles available at node (i,t), we define the following notation:

$$\tilde{V}_{it}(a) = V_{it} + e_a \tag{18}$$

where e_a is a unit vector with the only positive component in position a (that is, we assume we have one more vehicle of type a at node (i,t). Then the following indicator variables are computed:

$$\frac{\partial x_{a'lt}}{\partial V^+_{ait}} = X^+_{a'lt} = x_{a'lt}(\tilde{V}_{it}(a), \nu_{t+1}, u_{it}, \mathcal{L}_{it}) - x_{a'lt}(V_{it}, \nu_{t+1}, u_{it}, \mathcal{L}_{it}) \tag{19}$$

$$\frac{\partial y_{a'ijt}}{\partial V^+_{ait}} = Y^+_{a'ijt} = y_{a'ijt}(\tilde{V}_{it}(a), \nu_{t+1}, u_{it}, \mathcal{L}_{it}) - y_{a'ijt}(V_{it}, \nu_{t+1}, u_{it}, \mathcal{L}_{it}) \tag{20}$$

$$Z^+_{a'iit} = \tilde{y}_{a'iit}(\tilde{V}_{it}(a), \xi_{t+1}, u_{it}, \mathcal{L}_{it}) - \tilde{y}_{a'iit}(V_{it}, \xi_{t+1}, u_{it}, \mathcal{L}_{it}) \tag{21}$$

where $\tilde{y}$ represents the slack variables for equations (13).

It is possible that the additional vehicle in node (i,t) will satisfy a load that would otherwise be on hold, which results in an increase in the corresponding X variable. It is possible that it be repositioned to another location, which results in an increase in the corresponding Y variable. Finally, the increased marginal supply may be simply held in inventory, resulting in an increase in Z. Due to the multicommodity nature of the problem, there may also happen substitutions, resulting in increases or decreases in several X and Y variables. The right gradient is approximated by

$$\begin{aligned}\nu^+_{ait} \simeq & \sum_{j\in\mathcal{C}} \sum_{l\in\bar{\mathcal{L}}_{ijt}} \sum_{a'\in\mathcal{A}} X^+_{a'lt} \left(r_{a'lt} + \nu^+_{a',j,t+\tau_{a'ij}} + \sum_{t'>t} \sum_{a''\in\mathcal{A}} \hat{x}_{a''lt'}(-r_{a''lt'} + \nu^+_{a'',i,t'} - \nu^-_{a'',j,t'+\tau_{a''ij}}) \right) \\ & + \sum_{j\in\mathcal{C}} \sum_{a'\in\mathcal{A}} Y^+_{a'ijt} \left(-c_{a'ij} + \nu^+_{a',j,t+\tau_{a'ij}} \right) + \sum_{a'\in\mathcal{A}} Z^+_{a'iit}\, \nu^+_{a',i,t+1}\end{aligned} \tag{22}$$

where

$$\hat{x}_{a''lt'} = x_{a''lt'}(V_{i,t'}, \nu_{t'+1}, u_{i,t'}, \mathcal{L}_{it'}) \tag{23}$$

Equation (22) has three components. The first one captures the perturbations on loaded movements at time t, and at times $t' > t$. The second captures perturbations from changes in the repositioning decisions and the third changes from inventory decisions.

The left gradient is computed in the same fashion, but it is only defined where $V_{ait} > 0$. We define the following notation:

$$\bar{V}_{it}(a) = V_{it} - e_a \tag{24}$$

Then compute the following indicator variables:

$$\frac{\partial x_{a'lt}}{\partial V^-_{ait}} = X^-_{a'lt} = x_{a'lt}(V_{it}, \xi_{t+1}, u_{it}, \mathcal{L}_{it}) - x_{a'lt}(\bar{V}_{it}(a), \xi_{t+1}, u_{it}, \mathcal{L}_{it}) \tag{25}$$

$$\frac{\partial y_{a'ijt}}{\partial V^-_{ait}} = Y^-_{a'ijt} = y_{a'ijt}(V_{it}, \xi_{t+1}, u_{it}, \mathcal{L}_{it}) - y_{a'ijt}(\bar{V}_{it}(a), \xi_{t+1}, u_{it}, \mathcal{L}_{it}) \tag{26}$$

$$Z^-_{a'iit} = \tilde{y}_{a'iit}(V_{it}, \xi_{t+1}, u_{it}, \mathcal{L}_{it}) - \tilde{y}_{a'iit}(\bar{V}_{it}(a), \xi_{t+1}, u_{it}, \mathcal{L}_{it}) \tag{27}$$

The left gradient is approximated by

$$\begin{aligned}\nu^-_{ait} \simeq & \sum_{j\in\mathcal{C}} \sum_{l\in\bar{\mathcal{L}}_{ijt}} \sum_{a'\in\mathcal{A}} X^-_{a'lt} \left(r_{a'lt} + \nu^-_{a',j,t+\tau_{a'ij}} + \sum_{t'>t} \sum_{a''\in\mathcal{A}} \hat{x}_{a''lt'}(-r_{a''lt'} + \nu^-_{a'',i,t'} - \nu^+_{a'',j,t'+\tau_{a''ij}}) \right) \\ & + \sum_{j\in\mathcal{C}} Y^-_{a'ijt} \left(-c_{a'ij} + \nu^-_{a',j,t+\tau_{a'ij}} \right) + Z^-_{a'iit} \nu^-_{a',i,t+1}\end{aligned} \tag{28}$$

The calculation is basically the same as the right gradient, except that we use the left derivatives for x and y (equations (25), (26) and (27)), and we use the left derivatives ν^- reflecting that we are coming from the left rather than the right side.

Control Adjustment: The gradients computed in the backward pass of iteration n are used to update the profit potential function for iteration $n+1$ according with:

$$\xi^{n+1}_{ait} = \gamma \nu^n_{ait} + (1-\gamma)\xi^n_{ait} \tag{29}$$

where γ is the smoothing factor such that $0 < \gamma \leq 1$.

The gradients are also used to update the upper bounds. We define the following:

$$\eta^+_{aijt'} = \begin{cases} -c_{aij} + \nu^+_{a,j,t'+1} - \nu^-_{ait'} & \text{if } -c_{aij} + \nu^+_{a,j,t'+1} - \nu^-_{ait'} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

To measure the value of decreasing an upper bound, we define:

$$\eta^-_{aijt'} = \begin{cases} c_{aij} - \nu^-_{a,j,t'+1} + \nu^+_{ait'} & \text{if } c_{aij} - \nu^-_{a,j,t'+1} + \nu^+_{ait'} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

It is shown in Powell & Carvalho (1997*a*) that $\eta^+_{aijt'}$ and $\eta^-_{aijt'}$ are actually approximations to the right and left gradients of $\hat{G}$ with respect to $u_{aijt'}$. The upper bound adjustment consists of finding the highest value among all η^+_{aijt} and η^-_{aijt}. In case the highest value is a member of η^+, the corresponding upper bound is increased by one unit, otherwise it is decreased by one unit.

In section 4 we return to the LQN methodology to highlight the differences between the standard multicommodity model and the ROE model. In section 5 we build upon the standard multicommodity model to describe the flatcar model.

4 The Railroad Owned Equipment Model

This model is very similar to the multicommodity LQN presented in section 3. The major difference is that once a trailer is assigned to satisfy a request, it vanishes from the network, instead of appearing at some other terminal. Requests for empty trailers can be satisfied by assigning an empty trailer from the stock at the terminal the request originates. Thus, the contribution to the objective function for satisfying a request is the resulting profit. Based on the value of empty trailers at other terminals, empty trailers can also be assigned to be repositioned, resulting in a contribution to the objective function equivalent to the negative of the repositioning cost.

The the LQN algorithm for the ROE model is composed of the same three steps given in figure 5 that run iteratively. We take advantage of the notation defined in section 3 to highlight the differences between the ROE model and the multicommodity model presented in that section. We now provide a brief description of particular features of this model.

Forward Pass : This step finds a feasible solution for the problem of assigning empty trailers to requests over the entire horizon. It consists of solving *the local problem* for each terminal, for each time period within the planning horizon. For the ROE model, the objective function of the local problem (10) is replaced by

$$\max_{x_t, y_t} \sum_{a \in \mathcal{A}} \left(\sum_{b \in \mathcal{B}} \sum_{l \in \mathcal{L}^b_{it}} r_{alt} x_{alt} - \sum_{j \in \mathcal{C}} (c_{aij} - \xi_{a,j,t+\tau_{ij}}) y_{aijt} \right) \tag{32}$$

Backward Pass : Notice that, as opposed to the model in section 3, trailers are assigned to requests and then disappear from the network. Therefore, in the gradient computation, there

is no downstream effect of having an additional trailer satisfying a demand at any given node. Downstream effects only appear if the trailer is assigned to move to another terminal or is kept in inventory. This simplifies the backward pass considerably.

Equation (22) is replaced by

$$\begin{aligned}\nu^+_{ait} \;\simeq\; & \sum_{l\in\bar{\mathcal{L}}_{it}}\sum_{a'\in\mathcal{A}} X^+_{a'lt}\left(r_{a'lt}+\sum_{t'>t}\sum_{a''\in\mathcal{A}}\hat{x}_{a''lt'}(-r_{a''lt'}+\nu^+_{a''it'})\right)+\\ & \sum_{j\in\mathcal{C}}\sum_{a'\in\mathcal{A}} Y^+_{a'ijt}\left(-c_{a'ij}+\nu^+_{a',j,t+\tau_{a'ij}}\right)\end{aligned} \tag{33}$$

where

$$\hat{x}_{a''lt'} = x_{a''lt'}(V_{it'},\nu_{t'+1},u_{it'},\mathcal{L}_{it'}) \tag{34}$$

and equation (28) is replaced by

$$\begin{aligned}\nu^-_{ait} \;\simeq\; & \sum_{l\in\bar{\mathcal{L}}_{it}}\sum_{a'\in\mathcal{A}} X^-_{a'lt}\left(r_{a'lt}+\sum_{t'>t}\sum_{a''\in\mathcal{A}}\frac{\partial x_{a''lt'}}{\partial V_{a'it}}(-r_{a''lt'}+\nu^-_{a''it'})\right)+\\ & \sum_{j\in\mathcal{C}} Y^-_{a'ijt}\left(-c_{a'ij}+\nu^-_{a',j,t+\tau_{a'ij}}\right)\end{aligned} \tag{35}$$

No customization is needed for the control adjustment.

5 Flatcar Model

To describe the model for the flatcar problem, we take advantage of the notation defined in section 3 and adapt it to describe the flatcar model. Thus, in this section

- $\mathcal{A}$ is the set of flatcar types.
- $\mathcal{K}$ is the set of boxes k, real and forecasted, of all types, within the planning horizon, T. The set $\mathcal{K}$ plays the same role as the set $\mathcal{L}$ does in the basic LQN formulation, containing the set of loads that need to be moved.
- $\mathcal{T}_k$ is the set of feasible times for moving box $k \in \mathcal{K}$.
- $\bar{\mathcal{K}}_{it}$ is the set of boxes $k \in \mathcal{K}$ at terminal i available to be moved at time t.

- $\mathcal{K}^0_{it}$ is the set of boxes k at terminal i, becoming available at time t.
- $\mathcal{K}^s_{it}$ is the set of boxes k at terminal i, departing at time t.
- R_{ait} is the net number of flatcars type a becoming available at terminal i at time t.
- r_{akt} is the profit generated by choosing flatcar type a to move box k at time t.
- c_{aij} is the cost of moving flatcar type a over link (i, j, t).
- V_{ait} is the number of flatcars type a available at node (i, t).

To represent the feasible combinations of boxes that can go on each flatcar, we take advantage of the fact that larger flatcars may be divided into slots which can be independently filled. Most often, each slot will take one box. The maximum number of boxes any slot could possibly take is three in the event double stacking is allowed. In our particular problem, we divided the boxes in 28 types. Slots that take one box can therefore be filled at most in 28 manners, usually in less than 10 due to width and length constraints and the presence or absence of a hitch. The largest number of ways to fill a slot that takes three boxes is 80. We generate and sort the elements of the list of feasible combinations for each slot type in advance, considering the physical limitations of each slot (width, length, presence of hitches, refrigeration and ability to handle double–stacking). We define $\mathcal{F}_{as}$ as the set of feasible combinations for a given slot s in a flatcar type a.

For this model we have the following decision variables:

- $x_{akt} = 1$ if box k is moved on a flatcar type a starting at time t.
- w_{aijt} is the total number of flatcars of type a being moved along link (i, j, t).

The objective function is composed of the sum of the contributions of the decisions taken at each terminal at each time period within the planning horizon.

$$F(x, w) = \sum_{t=0}^{T} \sum_{i \in \mathcal{C}} \sum_{a \in \mathcal{A}} \left(\sum_{l \in \mathcal{K}_{it}} r_{alt} x_{alt} - \sum_{j \in \mathcal{C}} c_{aij} w_{aijt} \right) \tag{36}$$

Using the LQN approach, we do not solve the whole problem at once, but solve several local problems that yield feasible values for the decision variables x and w. The decision variables that control the local problems are:

- ξ_{ait} is the spatial potential function for flatcars type a at terminal i at time t.
- u_{aijt} is the upper bound on the number of flatcars type a that can be moved empty from terminal i to terminal j starting at time t.

The objective function for the local problem for node (i,t) that must be maximized is:

$$f_{it}(x, w, V_{it}, u, \xi) = \sum_{a \in \mathcal{A}} \left(\sum_{l \in \bar{\mathcal{K}}_{it}} r_{alt} x_{alt} - \sum_{j \in \mathcal{C}} (c_{aij} + \xi_{a,j,t+\tau_{ij}}) w_{aijt} \right) \tag{37}$$

The local problem is subject to the following constraints. The upper bounds on moving empty flatcars must be obeyed:

$$y_{aijt} \leq u_{aijt} \;\; \forall\, a \in \mathcal{A}, \;\; \forall\, j \in \mathcal{C} \tag{38}$$

where y_{aijt} represents the number of flatcars of type a moving empty from terminal i to terminal j departing at time t.

Each demand available at node (i,t) can be satisfied only once:

$$\sum_{a \in \mathcal{A}} x_{alt} \;\; \leq \;\; 1 \qquad \forall\, l \in \bar{\mathcal{K}}_{it} \tag{39}$$

One cannot dispatch more flatcars than available:

$$\sum_{j \in \mathcal{C}} w_{aijt} \;\; \leq \;\; V_{ait} \;\; \forall\, a \in \mathcal{A} \tag{40}$$

All the assignments of boxes to flatcars must be feasible. The complexity of the consolidation procedure would result in a very large integer program, even for a single local consolidation problem at a given node. We use instead a greedy heuristic to find a near optimal integer solution for each local problem. This heuristic satisfies the three types of constraints mentioned above and produces feasible assignments.

The heuristic consists of creating a matrix $M = \{M_{a,j}\}$ where $M_{a,j}$ is the value of the best feasible assignment of boxes destined to terminal j to a flatcar type a. Let $\mathcal{K}_{ijt}$ be the set of boxes available at terminal i at time t bound to terminal j. Each element $M_{a,j}$ is calculated using:

$$M_{a,j} = \sum_{k \in \mathcal{K}_{ijt}} \tilde{x}_{akt}\, r_{akt} - c_{aij} + \xi_{a,j,t+\tau_{ij}} \tag{41}$$

where the vector $\tilde{x}$ is obtained by enumerating the feasible combinations of boxes on flatcar type a and $\tilde{x}_{akt} = 1$ if box k is in the combination with the highest fill and $\tilde{x}_{akt} = 0$ otherwise.

The first component in (41) is obtained by looking at all the combinations of boxes available at node (i,t) bound to terminal j that could possibly be transported by flatcar type a. This component gives the "profit" of the combination that has the highest fill (defined as sum of the lengths of the boxes in the combination) among all the feasible combinations of boxes. The second component is the transportation cost of moving a flatcar of type a from terminal i to terminal j. The first two components, then, give the direct "contribution" of a particular assignment of boxes to a flatcar. The third component uses the spatial potential function to estimate the value of the flatcar at the destination. Without this component, we would have a standard, myopic model. Later, we will compare local decision making without network effects to local decision making with network effects by simply excluding or including the third term in equation (41).

The variable $\tilde{x}$ is a function of $\mathcal{K}_{ijt}$ and the flatcar type. Let L_k be the length of box k. Given the set $\mathcal{K}_{ijt}$ and the slot s, we find the combination of boxes that returns the highest fill by solving:

$$\tilde{x}_{sakt} = \begin{cases} 1 & \text{if } k \in \tilde{\mathcal{P}}, \text{ where } \tilde{\mathcal{P}}_{asj} = argmax\{\sum_{k' \in \mathcal{P}} L_{k'} \ \forall \mathcal{P}, \mathcal{P} \in \mathcal{F}_{as} \text{ and } \mathcal{P} \in \mathcal{K}_{ijt}\} \\ 0 & \text{otherwise} \end{cases} \tag{42}$$

After $\tilde{x}$ is computed for a given slot s, a temporary copy of $\mathcal{K}_{ijt}$ is updated by removing from $\mathcal{K}_{ijt}$ the elements from $\tilde{\mathcal{P}}$. After $\tilde{x}$ is computed for all the slots of a flatcar,

$$\tilde{x}_{akt} = \sum_{s} \tilde{x}_{sakt} \tag{43}$$

If no combination of boxes can be placed on a flatcar, the possibility of sending an empty flatcar of that type on that leg is considered. If $u_{aijt} > 0$, the empty trip is priced as

$$M^{e}_{a,j} = \xi_{a,j,t+\tau_{ij}} - c_{aij} \tag{44}$$

After all the elements of the matrix M have been computed, we select the highest valued element and validate its assignment. Let a' and j' be the indices of the largest element in M. Then:

$$x_{a'kt} = \tilde{x}_{a'kt} \ \forall k \in \tilde{\mathcal{P}}_{a's j'}, \ \forall s \tag{45}$$

$$w_{a'ij't} \leftarrow w_{a'ij't} + 1 \tag{46}$$

The matrix of assignment values must then be updated. One fewer flatcar of the type that has been selected will be available:

$$V_{a'it} \leftarrow V_{a'it} - 1 \tag{47}$$

Boxes bound to that destination that are contained in the chosen assignment become unavailable:

$$\mathcal{K}_{ijt} \leftarrow \mathcal{K}_{ijt} \setminus \tilde{\mathcal{P}}_{a'sj'} \tag{48}$$

The values in the matrix which are in the row and the column of the selected assignment need updating, which can be done by using equation (41). The procedure stops when all values in the matrix are *zero*. The result is a sorted list of tasks which are feasible and are implemented within the simulation.

This approach provides a greedy, heuristic solution to the problem of assigning boxes to flatcars. Its primary advantages are computational speed (we will need to solve this problem thousands of times in a run of our model) and the ability to explicitly consider any operational constraint which affects the ability to assign a set of boxes to a flatcar. Note that our solution is intended only to provide an estimate of the value of a particular type of flatcar at a particular terminal, at a point in time. A more optimal solution would likely be of little value since we do not have information on the location of flatcars within a yard, and the sequencing of flatcars on the train. This information is impractical to obtain at a network level at a given instant in time. More importantly, it is not even available when we are forecasting days or weeks into the future.

We now provide the description of the forward pass, backward pass and control adjustment for the flatcar model.

Forward Pass When assigning boxes to flatcars, one must be aware that intermodal trains run on a fixed schedule. This helps computation times, as it reduces the number of possible departure times for each box. Instead of having to solve one assignment problem for each time period at each terminal, the assignment problem needs to be solved only before a scheduled departure at each terminal. For our particular problem, this reduces the number of local problems that must solved at each iteration from about 850 to 250.

After all the local problems for a given time period are solved, the number of flatcars of each

type available at each terminal j is updated by

$$V_{a,j,t+1} \;=\; V_{ajt} - \sum_i w_{ajit} + \sum_i w_{a,i,j,t+1-\tau_{ij}} + R_{a,i,t+1} \;\; \forall j \in \mathcal{C},\; \forall a \in \mathcal{A} \tag{49}$$

Also, the set of boxes available at each terminal is updated using

$$\bar{\mathcal{K}}_{j,t+1} \;=\; \{\bar{\mathcal{K}_{jt}} \setminus \mathcal{K}^s_{jt}\} \cup \mathcal{K}^0_{j,t+1} \quad \forall j \in \mathcal{C} \tag{50}$$

Backward Pass As in the multicommodity case, the solution obtained in the forward pass is used to compute approximations for the gradients of the objective function with respect to V_{ait}, for each flatcar type a whenever there is a scheduled departure at node (i,t). The approximation consists of computing a finite difference. An additional flatcar of type a is added to node (i,t). The local problem at (i,t) is resolved and the increase in the objective function value of the local problem is taken as the approximation for the gradient. As before (see equation (18)) we represent a perturbation to the supply of flatcars at node (i,t), using $\tilde{V}_{it}(a) = V_{it} + e_a$.

The right gradient is approximated by

$$\nu^+_{ait} \;\simeq\; f_{it}(x, w, \tilde{V}_{it}(a), u, \nu) - f_{it}(x, w, V_{it}, u, \nu) \tag{51}$$

The left gradient is approximated in the same fashion, but it is only defined where $V_{ait} > 0$. Again using $\bar{V}_{it}(a) = V_{it} - e_a$, we approximate the left gradient using:

$$\nu^-_{ait} \;\simeq\; f_{it}(x, w, V_{it}, u, \nu) - f_{it}(x, w, \bar{V}_{it}(a), u, \nu) \tag{52}$$

Control Adjustment The control adjustment follows the methodology outlined in section 3. Gradients of the objective function with respect to u are computed and used in a coordinate search procedure to update u.

At the end of iteration n, the spatial potential function is updated according to:

$$\xi^{n+1}_{ait} = \gamma \nu^{n+1}_{ait} + (1-\gamma)\xi^n_{ait} \tag{53}$$

where the value of the smoothing factor γ, $0 < \gamma \leq 1$ is chosen experimentally.

γ	0.1	0.2	0.3	0.4	0.6
G^*	1337	1334	1334	1331	1327
coverage	99.7%	99.1%	99.1%	99.0%	98.7%

Table 1: Total contribution (in thousands of dollars) and percentage of requests covered within the planning horizon in the ROE model for selected values of γ.

6 Numerical Experiments

In this section, we first describe the calibration of the ROE model, involving the choice of an appropriate smoothing factor and the total number of iterations that the ROE model should run. We then proceed to the experiments necessary to calibrate the flatcar model, which involve choosing an appropriate smoothing factor. In this part we also provide some insights into the evolution of the network within the planning horizon. In the third part of this section we investigate the value of network information in the flatcar model. We show that the LQN approach applied to the flatcar management problem returns solutions that are better than those returned by a myopic assignment solved over the same planning horizon. Note that we use network information in the planning of empties; it is only in the choice of which flatcar to use locally that we ignore downstream effects (this mimics actual rail operations, which uses an approximation of global network information to plan empties, but allows local yard supervisors to choose what flatcar to use for a particular set of boxes).

6.1 ROE Model Calibration

The railroad handles two to three thousand requests for trailers every week. The ROE model is very similar to the multicommodity LQN described in Powell & Carvalho (1997*b*), which found a gap of about 3.5 percent between the LQN solution and the optimal value of the linear relaxation of the equivalent linear program. In fact, for the ROE model we can expect an even tighter bound, as the problem is not as resource constrained as the typical data sets examined in Powell & Carvalho (1997*b*). The planning horizon of the ROE model must match the planning horizon of the flatcar model, which was chosen as 10 days with 4-hour time periods.

Preliminary testing showed that the objective function of the ROE model reaches stable values

γ	0.1	0.2	0.3	0.4	0.6
Obj. function	3555	3551	3555	3558	3557

Table 2: Total contribution (in thousands of dollars) in the flatcar model for selected values of γ.

within 300 iterations. Each iteration for data sets with 10-day planning horizons takes on average 4.0 seconds. If the ROE model is set to run in batch, 300 iterations can be run within 20 minutes using a 256 MHz processor.

Table 1 shows the value of the objective function for the ROE model for selected values of γ. As expected, better results are obtained with small values of the smoothing factor. This table also shows the percentage of requests covered within the planning horizon, which is close to 100 percent, indicating very good performance. We chose $\gamma = 0.1$ for the smoothing factor for the ROE model.

6.2 Flatcar Model Calibration

The flatcar data set used for calibration is a real snapshot of the rail system. It contains 2300 boxes that are available to be moved at the beginning of the planning horizon. The demand generator and the ROE model provide the data for the rest of the planning horizon, comprising more than 16000 boxes forecasted to become available within the 10-day period. There are 21 different types of flatcar in the system.

Preliminary experiments were run with data sets having planning horizons of 5, 10 and 20 days. We chose to fix the planning horizon at 10 days because it provides a balance between good solution quality and acceptable speed. Each time period is four hours long, resulting in a planning horizon of 60 time periods. As with the ROE model, the average cpu time to run each iteration is about four minutes on a 256 MHz processor. After initial testing, we decided to stop the flatcar model after 100 iterations, giving us run times on the order of 3.6 hours.

We again performed experiments to determine an appropriate value for the smoothing parameter γ. The results of these experiments are shown in table 2. These results indicate similar performance for values of γ over the entire range from 0.1 to 0.6, although we do not have a quick explanation for this unexpected behavior, since we have generally found that values of γ over 0.5 do not work as well as smaller values. We chose to use an intermediate value of $\gamma = 0.3$.

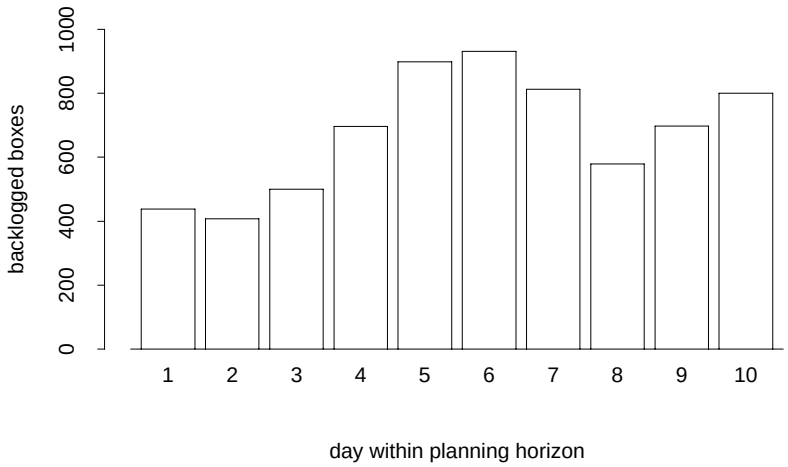

Figure 6: Number of backlogged boxes at the end of each day, after 100 iterations.

Figure 6 shows the total number of backlogged boxes at the end of each day after 100 iterations. We also show the number of flatcars that are moved empty each day in the network (figure 7). Flatcars that are dispatched partially filled are not included. For the calibration data set, day 1 corresponds to a Monday. When comparing days 1, 2, 3 to the same weekdays in the following week (days 8, 9 and 10), we can see truncation effects. The truncation of the horizon results in a greater number of boxes being backlogged on days 8, 9 and 10 because there is little incentive in the simulation to reposition a flatcar in the time periods close to the end of the planning horizon. Even though in practice the number of flatcars being repositioned should be fairly stable but lower over the weekends, this is not exactly what we observe because of truncation effects.

6.3 The Value of Network Information

Here we compare two kinds of runs. The first one consists of runs similar to the calibration run for the flatcar model, with the objective function of the local problem as in equation (32), which we repeat here:

$$f_{it}(z, w, N_{it}, u, \xi) = \sum_{a \in \mathcal{A}} \left(\sum_{l \in \bar{\mathcal{K}}_{it}} r_{alt} z_{alt} - \sum_{j \in \mathcal{C}} (c_{aij} - \xi_{a,j,t+\tau_{ij}}) w_{aijt} \right) \tag{54}$$

The second type of run is one where network information is not considered when solving the

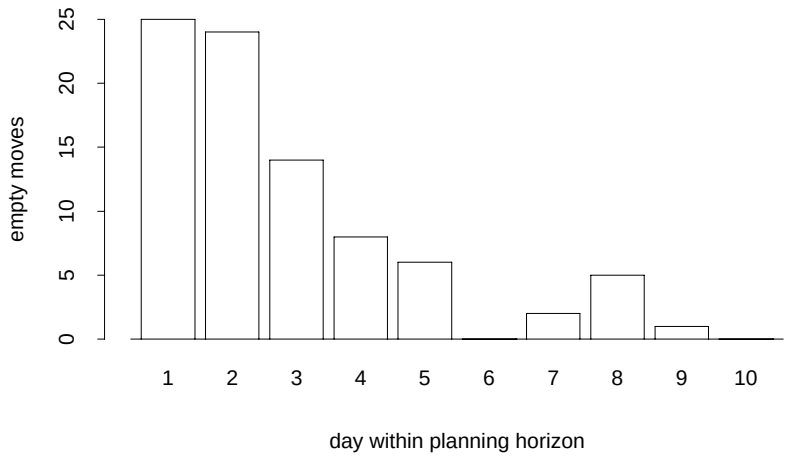

Figure 7: Number of empty flatcars being dispatched each day, after the hundredth iteration.

local problems. In this case, the objective function for the local problem is

$$f_{it}^{*}(z, w, N_{it}, u) = \sum_{a \in \mathcal{A}} \left(\sum_{l \in \bar{\mathcal{K}}_{it}} r_{alt} z_{alt} - \sum_{j \in \mathcal{C}} c_{aij} w_{aijt} \right) \tag{55}$$

Let F_{local} represent the value of the global objective function obtained by using equation (55) as the local problem and F_{global} be its value by using (54). In order to look at the value of network information, four other snapshots of the system were randomly generated using parameters obtained from a 3-year historical data file. Additionally, in each of these data sets we have altered the number of flatcars in the system. We wish to investigate the behavior of our system when the supply of resources varies. Thus we created four data sets with fleets that are 90, 95, 105 and 110 percent of the size of the fleet for our real snapshot. These data were developed by randomly generating new datasets of flatcars. For the smaller datasets, we kept each flatcar with a probability of .90 and .95, respectively. For the larger datasets, we kept each flatcar, and generated another one just like it with probability .05 and .10, respectively.

Table 3 shows the results for both runs for the real data set and the randomly generated ones with perturbed fleet size. Our expectation is that the data sets that have tighter supplies of resources result in a higher value for network information. That is indeed the fact. The relative improvement in our objective function rises from 3.1 percent using a fleet that is 105 percent of the base, to 4.7 percent using a fleet that is 90 percent of the base.

data set	F_{global}	F_{local}	gain %	load coverage %	
				global	local
90%	3502	3344	4.7	85.8	81.4
95%	3530	3399	3.9	86.8	83.4
std	3555	3431	3.6	88.3	84.2
105%	3562	3452	3.2	89.4	86.9
110%	3530	3423	3.1	90.0	87.5

Table 3: The value of network information in the flatcar model.

A more meaningful estimate of potential cost savings is the size of the fleet required to achieve the same level of coverage. Using only local information to choose flatcars, we achieve virtually the same level of utilization (86.9 percent) using a fleet that is 105 percent of the base as we do using global network information and a fleet that is 95 percent of the base. *These initial experiments suggest that a flatcar fleet that is managed locally with the benefit of our network information can achieve the same demand coverage with a 10 percent smaller fleet as compared to local management without the benefit of our network information.*

This dramatic conclusion needs to be qualified. The result assumes that flatcars can be chosen in a way that solves the local model given by equation (54). In practice, this problem will be subject to local constraints that will reduce the impact of network effects. Of course, changes in basic railroad operations could reduce the effect of these local constraints. On the other hand, our choice of a relatively short planning horizon (10 days) will underestimate total benefits, since network information will have almost no impact for the last two or three days of the simulation. A more careful study of the problem should include simulations where the model is solved on a rolling horizon basis. Such a study could run longer simulations and consider the effects of randomness on the overall quality of the solution. Our belief, however, is that local yard constraints will prove, in practice, to be the primary factor limiting real-world benefits.

7 Implementation Issues

The modeling framework proposed in this paper is designed to be used in a production setting. The major challenge limiting the use of global optimization models in railroads is the lack of quality, timely data in a central database. However, much of this data is available locally, although sometimes only as head knowledge (that is, the yard supervisor knows the information but it is not transferred to the computer). For example, we might prefer to send a particular flatcar type to Chicago, considering only global network issues. The yard supervisor, on the other hand, might realize that the particular flatcar in question is buried in the yard and would require a lot of jostling to retrieve the flatcar.

Our modeling approach uses two strategies to handle this problem. First, we provide local decision makers with the information to make better decisions from a network level, allowing him to choose among "good" alternatives (but using local judgment). Rather than telling the yard supervisor *what* to do, we tell him *why*, and allow him to take other issues into account. Second, we update the system in real–time, recognizing that the decision made locally is likely to be different than what the model is recommending. Both strategies are critical to providing a robust system that will work well in the presence of missing or inaccurate data. A theme of this approach might be called "global perspective, local control."

The LQN methodology lends itself to both strategies. The algorithm itself uses a decentralized control technology. Each terminal i is assumed to make decisions based on immediate costs plus the value of capacity at the destination, given by the spatial potential function. The optimizer is adaptive, allowing frequent updates to the data, independent of whether an "optimal" solution has been found. At the end of each forward pass, the system looks for updates to the data, that are passed to the system as transaction files. Our model considers five types of updates:

- Type 1: New flatcar becomes available.
- Type 2: Flatcar becomes unavailable.
- Type 3: Flatcar is dispatched.
- Type 4: Box is dispatched.
- Type 5: New box becomes available.

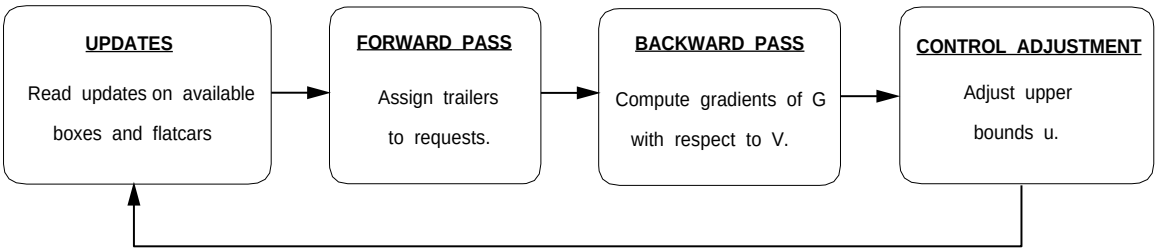

Figure 8: Flatcar LQN model.

The updates are fed to the model before each forward pass (figure 8). Once the data is updated, the model continues to optimize from the previous solution. As a result, new information is accepted by the model every few minutes, although it may take a few iterations for the new solution to settle down.

For the calibration data set in section 6, the objective function of the flatcar model reached stable values from a cold start after about 70 iterations. Thus, the startup time for the model to return estimates for the flow of flatcars and the spatial potential function is around five hours. Once the flatcar model has gone through the startup, it is possible to run it indefinitely and have up to date solutions every few minutes (for our problem, cycle times averaged around four minutes on a 133 MHz Silicon graphics workstation). Even if the process is stopped, the system can be restarted using the latest solution before the model is shut down. Thus, the five hour cold start occurs once and never needs to be repeated.

The decision variables are not used directly but support decisions made by dispatchers at each terminal. Gradient information is displayed to give the dispatcher the ability to weight decisions other than the ones suggested by the model. By supplying gradient information and the suggested flatcar flows, this model adds quantitative data to the qualitative knowledge decision makers have. In order to illustrate this, we have produced figure 9. This plot shows the spatial potential function for selected flatcar types and terminals at the beginning of the planning horizon.

These values were obtained after the model had run long enough to output stable solutions and thus suggested flatcar flows, including empty flatcars, have been factored in. The values reflect the state of the system according with our standard data and thus may change over time. One can start by noticing that in this particular example Norfolk is well supplied with flatcars, thus there is no value in sending in more flatcars. Table 4 provides a general idea of which boxes these

Figure 9: Spatial potential function for selected flatcar types at selected terminals.

flatcars can handle. Clearly, the more flexible flatcars tend to be more valuable. Type S151 is the only spine car type showing, and one can see it is more valuable at the terminals close to ports, Jacksonville and Memphis.

8 Conclusion

This paper documents the successful application of the LQN approach to a complex, real world problem. We believe the combined problem of flatcar and trailer management for intermodal operations cannot be modeled as a linear program without a high degree of simplification.

The LQN approach has room for considering a range of real world details, returns integer solutions and can be applied in a real time environment. It provides a new solution and can have its data base updated every few minutes.

Furthermore, in this paper we compare the overall contribution of the network over the 10–day planning horizon for two types of runs. We show that the gradient approximations provided by the LQN approach can be used to improve the total contribution by at least 3.0 percent when compared

flatcar type	trailer or container	range of box lengths	max number of boxes	double stacking
P534	T+C	0 - 57	2	NO
P312	T+C	0 - 48	1	NO
S311	T+C	40 - 53	1	NO
P533	T+C	40	2	NO
S310	C	48	2	YES
P310	T+C	40	1	NO
S151	C	40 - 48	10	YES

Table 4: General features of some flatcar types.

to a series of myopic local problems solved over the entire horizon. This improvement, of course, is dependent on several parameters of the problem, and thus may be higher or lower depending on the application.

It is interesting to notice that shortages of resources may happen not only when the number of flatcars in the system drops, but also when demand for service increases. Thus the benefit of network information on local decisions is the greatest during periods when the railroad may be having difficulty in meeting target deliveries.

References

Assad, A. (1987), 'Multicommodity network flows: A survey', *Networks* **8**(1), 37–92.

Chih, K. (1986), A real time dynamic optimal freight car management simulation model of the multiple railroad, multicommodity, temporal spatial network flow problem, Ph.D. Dissertation, Department of Civil Engineering and Operations Research, Princeton University.

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

Feeney, G. (1957), Controlling the distribution of empty cars, *in* 'Proc. 10th National Meeting, Operations Research Society of America'.

Feo, T. & Gonzaliez-Velarde, J. (1995), 'The intermodal trailer assignment problem', *Transportation Science* **29**, 330–341.

Godfrey, G. & Powell, W. (1995), Adaptive estimation of daily demands with complex calendar effects, Report 95-06, Department of Civil Engineering and Operations Research, Princeton University.

Herren, H. (1973), 'The distribution of empty wagons by means of computer: An analytical model for the Swiss Federal Railways (SSB)', *Rail International* **4**(1), 1005–1010.

Herren, H. (1977), 'Computer controlled empty wagon distribution on the SSB', *Rail International* **8**(1), 25–32.

Joborn, M. (1995), Empty freight car distribution at Swedish Railways - analysis and optimization modeling, Ph.D. thesis, Department of Mathematics, Linkoping University, Sweden.

Mendiratta, V. (1981), A dynamic optimization model of the empty car distribution process, Ph.D. Dissertation, Department of Civil Engineering, Northwestern University.

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

Powell, W. & Carvalho, T. (1997*a*), 'Dynamic control of logistics queueing network for large-scale fleet management', *Transportation Science*. (to appear).

Powell, W. & Carvalho, T. (1997*b*), 'Multicommodity logistics queueing networks', *European Journal of Operations Research*.

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

Shan, Y. (1985), A dynamic multicommodity network flow model for real-time optimal rail freight car management, Ph.D. dissertation, Princeton University.

Turnquist, M. (1986), Mov-em: A network optimization model for empty freight car distribution, School of Civil and Environmental Engineering, Cornell University.

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

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