Centralized and Decentralized Train Scheduling for Intermodal Operations

Alexandra M. Newman
Operations Research Department
Naval Postgraduate School
Monterey, CA 93943

Candace Arai Yano
Department of Industrial Engineering and Operations Research
University of California
Berkeley, CA 94720-1777

August 14, 2009

Centralized and Decentralized Train Scheduling for Intermodal Operations

Abstract:

We investigate a spectrum of decision-making approaches, from centralized to decentralized, within the context of scheduling direct and indirect (via a hub) trains and assigning containers to trains for the rail (linehaul) portion of the intermodal trip. The goal is to minimize operating costs, including a fixed charge for each train, variable transportation and handling costs for each container and yard storage costs, while meeting on-time delivery requirements. If shipping requirements are known, a centralized solution provides for better coordination, thereby reducing costs. However, information may be not available to support centralized decision-making. We present several methods for obtaining good solutions, and show that carefully-designed decentralized approaches may perform as well as centralized approaches for our problem.

1 Introduction

The question of determining the best method of using information and planning large-scale operations is not new. A wide variety of literature discusses the tradeoffs between using centralized vs. decentralized strategies considering organizational design, resource allocation and operational control.

A few authors focus on the degree to which decentralized information supports decision-making. Most of these models are concerned with (de)centralization of decision-making from an organizational perspective (e.g., Malone (1987), Malone and Smith (1988), Siebell (1990), Kendall and Schuldt (1993), Obel (1978), Anandalingam, Chatterjee and Gangolly (1987), Gazis (1987), and Leu, Rakes, Rees and Ceccucci (1992)).

Although optimization methods exist for coordinating decentralized decisions so as to achieve all or most of the benefits of centralized planning (e.g, Dantzig-Wolfe decomposition), relatively little research has been done to compare decentralized and centralized decision-making for operational problems (e.g., Chu, Lin, and Ng (1991), Kung and Marsden (1995), and Kouvelis and Gutierrez (1997)).

Our study differs from most of the prior research in that we are concerned with geographically (de)centralized decision-making in an environment where decisions are made over time. We allow operators in a subset of geographically distinct locations to make simultaneous decisions independently of each other, using only locally available data. This eliminates the need for systemwide data availability and a centralized planning system.

The remainder of the paper is organized as follows: Directly below, we describe our problem setting; then in Section 2, we present a formulation of the problem. In Section 3, we describe a range of decision-making strategies, from decentralized to centralized. Although our decentralized methods lead to optimization subproblems that are smaller than the original monolithic problem, some of these subproblems cannot be solved optimally for realistic instances. In Section 4, we present a heuristic preprocessing step that enables us to solve larger problems. We present numerical results in Section 5, and conclusions appear in Section 6.

Our problem setting closely parallels the train scheduling and container routing problem that we observed at the intermodal division of a Class I railroad. The company moves primarily sea cargo for major international shipping lines, as well as cargo for smaller local customers, from an intermodal terminal on the west coast through a hub in the west-central part of the U.S., and then to terminals east of the Mississippi River. The flow of traffic eastbound is greater than it is westbound, which is a common situation for U.S. railroads.

Intermodal retailers typically act as middlemen between the trucking and rail companies, reserve space on trains in advance, and then "sell" this space to their customers. These reservations contribute to the predictability of demand for the railroad. Overall, demand exhibits weekly patterns due to freighter schedules, and seasonal patterns due to factors such as traditional cycles in retail demand, and agricultural and manufacturing production.

Trains may be routed either directly to a destination, or through an intermediate terminal, where traffic originating at several locations but bound for a common destination may be consolidated onto a single train. This consolidation may cause up to several days' delay for transferring containers, repositioning railcars between trains, and waiting for the arrival of a train due to lack of coordination between inbound and outbound train schedules. We observed that decisions regarding whether trains should be sent directly or indirectly, and how the indirect trains should be coordinated at the hub are not made systematically. Furthermore, the train scheduling and container routing decisions do not appear to be affected strongly, if at all, by what speed of delivery has been promised, or what price has been charged to the customer. These observations motivated us to investigate how to schedule trains and route containers to achieve on-time delivery at minimum cost.

We address a short-term, finite-horizon, discrete-time scheduling problem for the rail linehaul portion of the intermodal trip. Given container demands differentiated by origin, destination, arrival date at origin, and due date at the destination, the objective is to determine a train schedule (for both direct and indirect trains) and container shipment plan to minimize the total cost over a horizon of a week to two weeks while meeting on-time delivery requirements and adhering to train capacity restrictions. The total cost consists of a fixed charge per limited-capacity train which depends on the transit time and locomotive requirements for the specific rail segment, a variable transportation cost per container which is proportional to the length of the rail segment, location-dependent container handling costs, and location-independent inventory holding (yard storage) costs for containers held at any terminal prior to their arrival at the destination. We assume that customers will accept delivery upon arrival at the destination, so no inventory is held at the destinations.

See Newman and Yano (1998) for a more detailed description of these costs.

We assume that hub delays and transit times are deterministic, constant across time, and that both are expressed as an integral number of time periods, where a time period is typically one day. Transit times are rarely predictable, but because time is expressed in days, not hours or minutes, there is implicit slack in the schedule to accommodate most unforeseen events. Explicit slack can be included (as is done in practice) to help ensure on-time delivery by further inflating scheduled transit times between terminals and scheduled delays at intermediate hubs.

We assume there is no limit on the number of trains that can be sent each day, although in reality, locomotive availability may be limited with respect to location and time. We further assume that the capacity of a train on each transportation segment is known, and that containers are homogeneous in terms of their use of train capacity. We consider the flow of trains and containers in one direction and do not address the repositioning of locomotives or empty containers. For further discussion of these issues see Folk and Bharadwaj (1980), Kikuchi (1985), Glickman and Sherali (1985), and Haghani (1989).

Research has been done on determining steady-state train frequencies over a horizon of weeks to months. Keaton (1989, 1992) addresses the problem of determining the frequency of service and the train type (i.e., direct or indirect) to be scheduled between terminals, the routing of cars on trains and through intermediate terminals, and blocking (or grouping of cars into shipment units). He models the problem as a mixed integer program and uses Lagrangian relaxation to solve it. Crainic and Rousseau (1986) develop an optimization framework for medium- to long-term service network planning for multimode freight transportation. Decisions include transportation modes and routes for various types of demands, service frequencies, consolidation and transfer policies at terminals, and the routing of freight. Their solution procedure relies on decomposition and column generation. Marín and Salmerón (1996) investigate the problem of determining a train schedule for a rail network, and the assignment of rail cars to these trains such that each train carries cars of a single service class. They employ simulated annealing and tabu search to solve the problem. The most significant difference between these models and our model is that the former determine only train frequencies based on aggregate demand rates, and not train timing, whereas our model provides detailed day-of-week train scheduling and container routing plans, and allows for time-varying demands.

The research on determining short-term day-of-week train schedules is sparse, and most of it is very recent. Barnhart and Ratliff (1993) develop a model that seeks to find the least expensive routings for a set of trailer movements by truck and/or rail on a single day, considering the possibility of transporting trailers from different sources on the same (rail) flatcar for an intermediate segment of the journey. Morlok and Peterson (1970) introduce a multi-period model to minimize the sum of fixed train and variable operating costs while adhering to due date restrictions for time-sensitive goods. Decisions are which trains to operate (differentiated by their departure time, routing, set of stops, and capacity) and which freight to assign to each train. The authors use branch-and-bound to solve a small instance of the resulting multicommodity network design problem. Nozick and Morlok (1997) address the rail movement of intermodal freight within the context of rail-truck intermodal transportation given a fixed train schedule over a finite horizon, taking equipment and locomotive repositioning into account. The objective is to minimize the cost of delivery such that the movements are feasible with respect to equipment availability, and the goods are delivered on time. Gorman (1998a, 1998b) considers a problem similar to ours: that of simultaneously establishing a train schedule and container routing to meet service requirements and yard, line, and train capacity restrictions. He employs a tabu-enhanced genetic search to arrive at solutions within 10% of the optimum for the special case of a single origin and single destination with multiple routes between them. He then uses his procedure to obtain a solution for one problem with multiple interdependent origins and destinations. Although the solution he obtains represents a significant improvement over current practice, a comparison with a bound or an exact solution is not provided.

Our model differs from prior work in that we simultaneously determine an explicit direct and indirect train schedule and the corresponding container routing decisions for multiple interdependent origins and destinations using a formal optimization approach. This approach enables us to assess the tradeoffs between computation time and the quality of solutions obtained from a centrally planned system and one in which various degrees of decentralized decision-making are allowed.

2 Mathematical Formulation and Problem Structure

Recall that our problem is to choose train schedules and container routes for each day in a horizon, so as to achieve on-time delivery at minimum cost. The problem is difficult because of: (i) the possibility of sending more than one train on each segment each day, (ii) the option of sending both direct and indirect trains, (iii) the dynamic arrival of containers, and (iv) distinct due dates for different customer orders.

The subscripts in the model are as follows:

i = index of origins, i = 1,...,I

j = index of hubs, j = 1,...,J

k = index of destinations, k = 1,...,K

t = index of days in the time horizon

l = index of level of service, i.e., the due date of the container at the destination

The parameters in the model are as follows:

α_{ik} = direct transportation time between origin i and destination k

β_{ij} = transportation time between origin i and hub j

γ_{jk} = transportation time between hub j and destination k

δ_j = delay time incurred from passing through hub j

C = capacity of a train (number of containers)

h = holding cost of a container ($/container/day)

c^a_{ik} = variable unit cost of transporting a container directly from origin i to destination k

c^e_{ijk} = variable unit cost of transporting a container from origin i via hub j to destination k

S^{ao}_{ik} = fixed cost of running a train directly between origin i and destination k, including labor and train assembly

S^{eo}_{ij} = fixed cost of running a train between origin i and hub j, including labor and train assembly

S^{h}_{jk} = fixed cost of running a train between hub j and destination k, including labor and train assembly

g^{o}_{i} = cost of placing a container on the train at origin i

g^{h}_{j} = cost of rearranging a container at hub j

g^{d}_{k} = cost of removing a container from the train at destination k

b_{iktl} = the number of containers that arrive at origin i on day t bound for destination k with a due date of time l

The decision variables are as follows:

I^{o}_{iktl} = container inventory held at origin i at time t, which is due at destination k by time l

I^{h}_{ijktl} = container inventory originating at i and held at hub j at time t, which is due at destination k by time l

I^{d}_{iktl} = container inventory from origin i due by time l which is held at destination k at time t

x^{ao}_{iktl} = number of containers shipped directly from origin i to destination k at time t, which are due by time l

x^{eo}_{ijktl} = number of containers shipped from origin i to hub j at time t, which are due at destination k by time l

x^{h}_{ijktl} = number of containers which originated at i and are shipped at time t from hub j to destination k, where they are due by time l

z^{ao}_{ikt} = the number of trains sent directly from origin i to destination k at time t

z^{eo}_{ijt} = number of trains sent from origin i to hub j at time t

z^{h}_{jkt} = number of trains sent from hub j to destination k at time t

The formulation follows:

$$(P) : min \quad Z =$$

$$\sum_{iktl} h * I^o_{iktl} + \sum_{ijktl} h * I^h_{ijktl} + \sum_{ijkl} \sum_{w=t+\beta_{ij}}^{t+\beta_{ij}+\delta_j} h * x^{eo}_{ijkwl} + \sum_{iktl} c^a_{ik} * x^{ao}_{iktl} + \sum_{ijktl} c^e_{ijk} * x^{eo}_{ijktl}$$
$$+ \sum_{iktl} g^o_i * x^{ao}_{iktl} + \sum_{ijktl} g^o_i * x^{eo}_{ijktl} + \sum_{ijktl} g^h_j * x^h_{ijktl} + \sum_{iktl} g^d_k * x^{ao}_{iktl} + \sum_{ijktl} g^d_k * x^h_{ijktl}$$
$$+ \sum_{ikt} S^{ao}_{ik} * z^{ao}_{ikt} + \sum_{ijt} S^{eo}_{ij} * z^{eo}_{ijt} + \sum_{jkt} S^h_{jk} * z^h_{jkt}$$

subject to

$$b_{iktl} + I^o_{ik(t-1)l} = I^o_{iktl} + x^{ao}_{iktl} + \sum_j x^{eo}_{ijktl} \qquad \forall i,k,t,l \tag{1}$$

$$I^h_{ijk(t-1)l} + x^{eo}_{ijk(t-\beta_{ij}-\delta_j)l} = I^h_{ijktl} + x^h_{ijktl} \qquad \forall i,j,k,t \ni t \geq 1 + \beta_{ij} + \delta_j, l \tag{2}$$

$$I^d_{ik(t-1)l} + x^{ao}_{ik(t-\alpha_{ik})l} + \sum_j x^h_{ijk(t-\gamma_{jk})l} = I^d_{iktl} + b_{iktl} \qquad \forall i,k,t \ni t \geq 1 + \alpha_{ik}, l \tag{3}$$

$$\sum_l x^{ao}_{iktl} \leq C * z^{ao}_{ikt} \quad \forall i,k,t \tag{4}$$

$$\sum_{kl} x^{eo}_{ijktl} \leq C * z^{eo}_{ijt} \quad \forall i,j,t \tag{5}$$

$$\sum_{il} x^h_{ijktl} \leq C * z^h_{jkt} \quad \forall j,k,t \tag{6}$$

$$I^o_{iktl},\ x^{ao}_{iktl},\ I^d_{iktl}\ \geq 0\ and\ integer \quad \forall i,k,t,l \tag{7}$$

$$I^h_{ijktl},\ x^{eo}_{ijktl},\ x^h_{ijktl}\ \geq 0\ and\ integer \quad \forall i,j,k,t,l \tag{8}$$

$$z^{ao}_{ikt}\ \geq 0\ and\ integer \quad \forall i,k,t \tag{9}$$

$$z^{eo}_{ijt}\ \geq 0\ and\ integer \quad \forall i,j,t \tag{10}$$

$$z^h_{jkt}\ \geq 0\ and\ integer \quad \forall j,k,t \tag{11}$$

where I^o_{iktl} is set equal to 0 unless $t \geq 1$ and $l > t + \alpha_{ik}$, I^h_{ijktl} is set equal to 0 unless $t \geq 1 + \beta_{ij} + \delta_j$ and $l > t + \gamma_{jk}$, and I^d_{iktl} is set equal to 0 unless $t \geq 1 + \alpha_{ik}$ and $l > t$. Note also that $x^{eo}_{ijktl} = 0$ if $l < t + \beta_{ij} + \gamma_{jk} + \delta_j$ and $x^h_{ijktl} = 0$ if $l < t + \gamma_{jk}$.

Although the formulation is written for the case in which there may be multiple hubs and each container may pass through at most one hub, in our analysis, we assume there is only one hub. We also assume that direct travel time between an origin and a destination is strictly less than the total transit and delay time for a container shipped indirectly, i.e., $\alpha_{ik} < \beta_{ij} + \gamma_{jk} + \delta_j$.

The objective function contains the following terms: the inventory holding cost at the origin and at the hub; the transportation cost of directly and indirectly shipped goods; the handling cost at the origin for both direct and indirect shipments, the handling cost at the hub for indirect shipments, and the handling cost at the destination for direct and indirect shipments; and finally, the fixed cost at the origin for direct trains and trains bound for a hub, and the fixed cost at the hub for indirect trains.

Constraints (1), (2), and (3) represent conservation of flow of containers at the origin, hub, and destination nodes, respectively. Constraints (4) require that for all origins, destinations, and time periods, the number of containers sent on direct trains must not exceed the total capacity of the trains departing. Likewise, constraints (5) and (6) ensure that train capacity is not exceeded on trains bound for the hub and trains leaving the hub, respectively. Finally, nonnegativity and integrality restrictions are imposed on all decision variables. Appropriate inventory variables are initialized to zero; others are constrained to be non-negative. Similarly, indirect container shipments are constrained to be zero if shipment due dates necessitate direct routing.

Our problem is modeled as a piecewise-concave-cost multicommodity network flow problem. Problems of this type are both theoretically and computationally difficult to solve. Even the fixed-charge (uncapacitated) multicommodity flow problem, which has a simpler structure than our problem, has been shown to be NP-complete (Garey and Johnson, 1979). Earlier work has examined special cases of the fixed-charge problem. Specifically, Palekar, Karwan, and Zionts (1987), and Lamar, Sheffi, and Powell (1990) examine methods to improve the performance of the fixed charge transportation problem. Schaffer and O'Leary (1989) treat a special case of the fixed-charge problem in which the fixed charges are associated with the supply point. Hochbaum and Segev (1994) and Herrmann, Ioannou, Proth, and Minis (1996) propose solution procedures for the general (single commodity) fixed-charge problem. Aggarwal, Oblak, and Vermuganti (1994) and Barnhart, Hane, and Vance (1996) treat multi-commodity flow problems with fixed upper bounds on the arc capacities. The constraints in these classes of problems differ from ours because our arc capacities depend on the number of trains sent, *and* the capacity is shared among multiple commodities.

Figure 1 depicts our network for two origins, two destinations and two time periods, assuming, for simplicity, instantaneous transit times and no delays at the hub. The network contains three sets of nodes, in addition to a source and sink. The first set consists of one node for each (origin, time period) pair. In the second set, there is one node for each time period, and the physical location is considered to be the hub. In the third set, there is one node for each (destination, time period) pair. An arc links two nodes if a container may travel from the location and time period associated with one node to those associated with the adjacent node. Arcs from a location in one time period to the same location in the subsequent time period permit inventory flows from period to period. The multicommodity nature of the problem manifests itself in that direct shipments on the same arc may have different due dates. Likewise, indirect shipments from an origin to a hub may have different destinations and/or due dates. Finally, indirect shipments from the hub to a destination may have different origins and/or different due dates. A variable cost is associated with transporting a specific commodity on a given arc. A commodity is assigned an infinite cost if its due date prohibits it from traveling along an arc at a given time period. The upper bound on the total flow on each arc depends on the number of trains scheduled to travel between the two locations in a given time period, and a fixed charge is assessed for each of these trains.

A centralized planner would face the problem described and formulated above. Typical problem instances contain thousands of general integer variables and thousands of constraints. Although some very small problems can be solved optimally in a matter of seconds or minutes, realistic problems present computational challenges. In practice, the problem is solved in a geographically decentralized manner with each terminal responding to forecasted container arrivals. In the next section, we present a variety of decision-making strategies with varying degrees of decentralization which afford, on the average, better quality solutions much more quickly than the centralized approach.

3 Decision-Making Strategies: From Decentralized to Centralized

There are many ways in which one can construct good, feasible solutions to the train scheduling and container routing problem. We describe each of these methods in turn, providing abbreviated formulations where appropriate. We first introduce the simplest and most decentralized approach on our spectrum of alternatives, *decentralized scheduling and routing*. For each origin, we solve the problem of how to schedule direct and indirect trains and route containers outbound from the origin, using adjusted costs that include an estimate of consequent costs incurred at the hub and between the hub and the final destinations. We refer to each of these as an *origin scheduling subproblem*. Then, given the resultant container arrivals at the hub, the train and container schedules outbound from the hub to each destination are optimized. We refer to each of these as a *hub scheduling subproblem*. We call this approach *decentralized* because each set of train scheduling and container routing decisions can be made locally either at each separate origin or at each hub, i.e., in a geographically decentralized way *without* the need to coordinate decisions with other locations and without the intervention of a central planner.

In each origin scheduling subproblem (which must be solved for all but the centralized solution approach), we assume that for each train inbound to the hub from origin i, there is a train outbound from the hub whose fixed charge is the average of the fixed charges incurred on all rail segments outbound from the hub to the various destinations. (If the demands differ widely by destination, one could instead use an appropriate weighted average of the fixed charges.) Also, the handling cost per container is the sum of the handling cost at the origin and at the hub. Thus, rather than using fixed charges and handling costs that reflect only the first transportation segment, we use adjusted costs that reflect estimates for the entire route:

$$\widetilde{S_{ij}^{eo}} = S_{ij}^{eo} + S_j^h$$

$$\widetilde{g_{ij}^{o}} = g_i^o + g_j^h$$

where S_j^h is the fixed cost at hub j, obtained by using an average or a weighted average of S_{jk}^h across destinations.

Note that because we assume that all containers incur variable costs proportional to the distance they are hauled, and that direct and indirect routes are virtually identical, transportation costs are sunk and equal, regardless of whether containers travel on a direct or an indirect route. To accurately account for transportation costs for indirectly routed containers in each origin scheduling subproblem, we assess the same variable cost for all containers regardless of whether they are transported directly or indirectly.

Our motivation for making these cost adjustments is to incorporate the first-order effects of sending trains from the origin to the hub on the costs that are incurred after the train reaches the hub. In effect, the cost adjustment is an estimate of the "cost to go" outbound from the hub. Although the number of trains into and out of the hub may not be exactly equal within a short time horizon, in practical applications these values are fairly well balanced. If the train capacities are well utilized inbound to the hub, on-time delivery requirements make it difficult to hold containers at the hub long enough to achieve significant additional consolidation outbound from the hub.

In solving any subproblem for any of the methods described in this section, all relevant constraints apply. For example, in solving each origin scheduling subproblem, we require conservation of flow of containers at the origin and satisfaction of train capacity constraints on both direct and indirect trains leaving the origin. For each hub scheduling subproblem, we require conservation of flow of the containers at the hub, and adherence to capacity constraints for all trains outbound from the hub. In both of these problems, containers must be shipped far enough in advance to arrive at the destination on time. Formulations of origin and hub scheduling subproblems for a single origin and for a single hub-destination pair, respectively, are given in Appendix A. Note that the solution and objective function value for the entire *origin scheduling problem* can be obtained by solving an origin scheduling *subproblem*, P_o^i, for each origin i, adding the objective function values, and taking the union of the solutions. Equivalently, were it tractable, the origin scheduling problem could be solved as a monolith, in which the objective function is summed across all origins, and the constraint set is the union of constraints imposed at all origins. Analogous reasoning holds for equivalently solving the independent hub scheduling subproblems for each hub-destination pair as a single *hub scheduling problem*. Therefore, for ease of exposition, in the ensuing discussion, we refer to the origin and hub scheduling problems as monoliths, and assume that they have been solved in a decoupled fashion to preserve computational tractability.

Let Z represent the objective function value for the original problem, and Z^* denote the optimal objective function value. Also let:

Z_1= the objective function value for the origin scheduling problem with the original cost parameters S_{ij}^{eo} and g_i^o

$\widetilde{Z_1}$= the objective function value for the origin scheduling problem with the adjusted cost parameters, $\widetilde{S_{ij}^{eo}}$ and $\widetilde{g_{ij}^o}$

Z_2= the objective function value for the hub scheduling problem

Let $\mathbf{x_i^o}$ and $\mathbf{x_j^h}$ be shorthand notation for all container flows from origin i and hub j, respectively:

$$\mathbf{x_i^o} = \{x_{iktl}^{ao},\ \ x_{ijktl}^{eo}\ \forall j,k,t,l\} \quad \forall i\ \ and\ \ \mathbf{x^o} = \{\mathbf{x_i^o},\ i = 1,...,I\}$$

$$\mathbf{x_j^h} = \{x_{ijktl}^{h}\ \forall i,k,t,l\} \quad \forall j\ \ and\ \ \mathbf{x^h} = \{\mathbf{x_j^h},\ j = 1,...,J\}$$

Finally, let $\mathbf{z_i^o}$ and $\mathbf{z_j^h}$ denote the trains outbound from origin i and hub j, respectively:

$$\mathbf{z_i^o} = \{z_{ikt}^{ao},\ \ z_{ijt}^{eo}\ \ \forall j,k,t\} \quad \forall i$$

$$\mathbf{z_j^h} = \{z_{jkt}^{h}\ \forall k,t\} \quad \forall j$$

Using the above notation, the *decentralized scheduling and routing* approach entails solving the following optimization problems:

$$1.\ \ Solve:\ \ \min_{\mathbf{x_i^o},\mathbf{z_i^o}} \widetilde{Z_1}(\mathbf{x_i^o},\mathbf{z_i^o})\ \ \forall i$$

$$2.\ \ Then,\ \ given\ \ \mathbf{x_i^o},\ i=1,...,I\ \ from\ \ step\ \ 1,\ \ solve:\ \ \min_{\mathbf{x_j^h},\mathbf{z_j^h}} Z_2(\mathbf{x_j^h},\mathbf{z_j^h}|\mathbf{x_i^o})\ \ \forall j$$

$$3.\ \ Finally,\ \ set\ \ Z^{(1)} = \sum_i Z_1(\mathbf{x_i^o},\mathbf{z_i^o}) + \sum_j Z_2(\mathbf{x_j^h},\mathbf{z_j^h})$$

where $Z^{(1)}$ denotes the objective function value for the monolithic problem when solved with the *decentralized scheduling and routing* approach.

The second variation is similar to the *decentralized scheduling and routing* approach, except that an intermediate step is interjected after the origin scheduling problem is solved. Recall that the origin scheduling problem uses only simple estimates of the "cost to go," and does not account for detailed scheduling of trains at the hub. In reality, the "cost to go" weakly decreases as containers arrive at the hub with more slack (providing more flexibility and more opportunity for consolidation at the hub), and as more containers are sent directly. Thus, while the origin scheduling problem may be indifferent among a variety of container allocations to the various scheduled trains, the actual costs incurred at the hub may differ widely. The purpose of the intermediate step is to make adjustments in the assignments of containers to trains in a systematic (but heuristic) way in order to improve the container routings and the train schedules outbound from the hub.

We make the adjustments by solving a variant of the container routing portion of the origin scheduling problem (with the same train schedule outbound from the origin as that found with the *decentralized scheduling and routing* approach). The form of the objective remains the same as in the origin scheduling problem. Among alternate optimal solutions, we wish to identify a container routing scheme that satisfies an earliest due date (EDD) sequence, taking into account the actual train schedule and differences in travel times between direct and indirect trains. To this end, we add constraints to ensure that for each origin-destination pair: (i) containers with a given due date are not shipped on a direct train unless all available containers with an earlier due date have been either shipped or already allocated to the direct train under consideration, and (ii) containers with a given due date are not shipped on an indirect train unless all containers with an earlier due date could be shipped, either on the current train, or on the next available direct train such that they arrive at the destination at least as early as if they had been allocated to the current train. (In Appendix B, we prove that if an alternate optimal solution to the origin scheduling problem exists, the objective value remains the same even when imposing these EDD constraints.) These constraints ensure that containers are sent in decreasing order of urgency and that containers arrive at the hub in EDD order (subject to their availability at the origin). We refer to this approach as the *decentralized scheduling and routing with* *intermediate container adjustment* approach. Observe that the container adjustments are made locally at each origin, so the decision process remains decentralized.

The third variation attempts to better coordinate container flows through the system by making all container routing decisions simultaneously after a train schedule is determined. The steps are as follows: Solve an origin scheduling subproblem for each origin. Solve a hub scheduling subproblem for each hub-destination pair using the container arrivals at the hub as determined by the origin scheduling problem. Fix the resulting train schedules outbound from the origin and outbound from the hub as determined from the origin and hub scheduling problems, respectively, and re-optimize all container movements. We term this the *decentralized scheduling with ex post routing* approach. Although the train schedules are determined using only local container arrival data, the container routing decisions require the entire (systemwide) train schedule. As such, the train schedules are determined locally, but a central authority must impose the container routing scheme.

Mathematically, this corresponds to the following optimization problems:

$$1.\quad Solve: \quad \min_{\mathbf{x_i^o}, \mathbf{z_i^o}} \widetilde{Z_1}(\mathbf{x_i^o}, \mathbf{z_i^o}) \quad \forall i$$

$$2.\quad Then,\ \ given\ \ \mathbf{x_i^o},\ i = 1, ..., I\ \ from\ \ step\ 1,\ \ solve: \quad \min_{\mathbf{x_j^h}, \mathbf{z_j^h}} Z_2(\mathbf{x_j^h}, \mathbf{z_j^h} | \mathbf{x_i^o}) \quad \forall j$$

$$3.\quad Finally,\ given\ \mathbf{z_i^o},\ i = 1, ..., I\ from\ \ step\ 1\ and\ \mathbf{z_j^h},\ j = 1, ..., J\ from\ step\ 2,\ \ solve:$$

$$Z^{(2)} = \min_{\mathbf{x^o}, \mathbf{x^h}} \{\sum_i Z_1(\mathbf{x_i^o} | \mathbf{z_i^o}) + \sum_j Z_2(\mathbf{x_j^h} | \mathbf{x^o}, \mathbf{z_j^h})\}$$

where $Z^{(2)}$ denotes the objective function value for the monolithic problem when solved with the *decentralized scheduling with ex post routing* approach.

The fourth approach is a variation of the third approach with an intermediate container adjustment after the solution for the origin scheduling problem has been determined. Recall that in the third approach, container flows are re-optimized after the train schedules have been established. As such, the reason for the intermediate container adjustment in this context is *not* to decide the exact container assignments, but to provide a pattern of container arrivals at the hub from which we can construct a low-cost train schedule outbound from the hub. Although adherence to an EDD shipping schedule outbound from the origins helps to reduce costs incurred outbound from the hub by increasing opportunities for consolidation, in preliminary tests, we found that adherence to the EDD schedule alone did not provide much benefit over the third approach (which is identical to this fourth approach but has no intermediate container adjustment). A little reflection will reveal that shipping additional containers on already-scheduled direct trains reduces the aggregate demand for trains outbound from the hub, and shipping containers as early as possible using available space on indirect trains inbound to the hub also contributes by improving consolidation opportunities. Our container adjustment scheme for the fourth approach takes these factors into account.

We again make the container adjustments by solving a variant of the container routing portion of the origin scheduling problem (using the train schedule outbound from the origin found with the *decentralized scheduling with ex post routing* approach). In addition to the terms in the standard origin scheduling problem (with a fixed train schedule), the objective also includes, for each origin-destination pair, large fixed-charge rewards for adhering to the EDD shipping sequence, as well as large fixed-charge rewards for each (due date, shipping period) pair for which all available containers have been completely shipped. The latter rewards tend to encourage shipment as soon as possible, to the extent space is available. Additional small (linear) incentives for shipping directly rather than indirectly are also included, which would, for example, encourage holding a container until the next day to ship it on a direct train rather than shipping it on an indirect train today. These latter incentives are small, however, so the fixed-charge rewards are the dominant influence. (We note that this second container adjustment method did not perform as well as the first one for the *decentralized scheduling and routing* approach, because the detailed container assignments are more critical in that method, which does not include the solution of a systemwide container routing problem at the end.)

The adjusted container flows are used as input to the hub scheduling problem. As in the *decentralized scheduling with ex post routing* approach, only the *train schedules* from the hub scheduling problem are fixed. Using the train schedule from the origin and hub scheduling problems, the systemwide container flow problem is solved. We refer to this approach as *decentralized scheduling using intermediate container adjustment and ex post routing.*

A fifth solution approach, which we refer to as *partially centralized scheduling and routing*, proceeds as follows. First, solve the origin scheduling problem for each origin. Then, fixing the resulting schedule of trains outbound from the origin, solve the problem of scheduling the trains outbound from the hub and *all* container movements (i.e., both those outbound from the origin and those outbound from the hub). This approach allows each origin to determine only its *train* schedule (and not the container flows) independently of the other origins. The remainder of the decisions are determined simultaneously by a central decision-maker, who requires train schedule and demand information at all origins. In other words, this is the most centralized of the decomposition schemes we have introduced.

Mathematically, this corresponds to the following optimization problems:

$$1. \quad Solve: \quad \min_{\mathbf{x_i^o}, \mathbf{z_i^o}} \widetilde{Z}_1(\mathbf{x_i^o}, \mathbf{z_i^o}) \quad \forall i$$

$$2. \quad Then, \;\; given \;\; \mathbf{z_i^o}, \; i = 1, ..., I \;\; from \;\; step \;\; 1, \;\; solve:$$

$$Z^{(3)} = \min_{\mathbf{x^o}} \{ \sum_i Z_1(\mathbf{x_i^o} | \mathbf{z_i^o}) + \sum_j \min_{\mathbf{x_j^h}, \mathbf{z_j^h}} Z_2(\mathbf{x_j^h}, \mathbf{z_j^h} | \mathbf{x^o}) \}$$

where $Z^{(3)}$ denotes the objective function value for the monolithic problem when solved with the *partially centralized scheduling and routing* approach.

The final method is the *centralized scheduling and routing* approach, which involves solving the entire monolithic problem. In this instance, all train scheduling and container routing decisions are made simultaneously. This is in contrast to the decomposition schemes we describe above in which at least a subset of the decisions are made locally. Note that the original (monolithic) problem can be posed as a nested optimization problem:

$$Z^* = \min_{\mathbf{z_i^o}, \, i=1,..,I} \{ \min_{\mathbf{x^o}} \{ \sum_i Z_1(\mathbf{x_i^o} | \mathbf{z_i^o}) + \sum_j \min_{\mathbf{x_j^h}, \mathbf{z_j^h}} Z_2(\mathbf{x_j^h}, \mathbf{z_j^h} | \mathbf{x^o}) \} \}$$

The inner problem is the hub scheduling problem, given container flows into the hub. The middle-level problem involves optimizing container flows outbound from the origin, taking into account both the costs incurred at the origin and the resulting (optimal) cost for the hub scheduling problem. The outer optimization problem establishes the train schedules outbound from the origins. The problem could be solved, in principle, using this approach, but would pose two formidable challenges. First, although the hub scheduling problem is easy to solve, the optimum cost for each destination is not a smooth function of the container arrivals. Thus, choosing the container flows in the middle-level problem is not easy. Second, choosing the train schedules outbound from the origins is a difficult combinatorial problem because of the partial substitutability of direct and indirect trains, and of trains scheduled at different times. It is also difficult because the impact of the train schedule at the origins on the middle (and inner) problem is indirect: it only defines constraints on certain container flows.

We do not solve the original problem using this nested optimization approach (instead, we use commercial optimization software), but the above formulation strongly motivated the various solution strategies outlined above. In essence, each of the first five solution approaches is an approximate method for solving the nested optimization problem.

We might expect to obtain better solutions from more centralized approaches, if the problems can be solved optimally or near-optimally. On the other hand, such approaches have several drawbacks. In problems with many locations and/or time periods, the necessary information may be difficult to gather and update. It also may be computationally burdensome or impossible to obtain good solutions when many decisions must be made simultaneously. Using one of the four *decentralized* methods allows us to decompose the hub scheduling problem by hub and destination, and each of these problems can be solved very efficiently (see Yano and Newman, 1998).

One important motivation for our particular approach for solving the origin scheduling problem is that, for an arbitrary train schedule outbound from the origin, the container routing portion of the origin scheduling problem can be represented as a single-commodity network flow model, which allows us to relax the integrality constraints on the container flows without loss of optimality. This contributes significantly to the computational efficiency of our procedures. However, as the number of destinations grows, it becomes increasingly more difficult to obtain optimal solutions to the origin scheduling subproblems (which must consider all destinations simultaneously). To deal with such situations, we have developed a variation of our decomposition method that relies on a preprocessing step to set certain direct train variables, as described in the next section.

4 Preprocessing Method to Determine Direct Trains for Many Destinations

Our method to obtain solutions for problems with a larger number of destinations relies on a heuristic preprocessing step to set the values of direct train variables in the origin scheduling subproblems. Our rationale for heuristically setting the direct train variables (to reduce the size of the remaining problem) is that these decisions depend primarily upon the demand between a single origin and a single destination, and are only indirectly affected by when and how containers are sent to other destinations. Moreover, the primary indirect effect can be captured largely in the flows of containers sent via the hub from the designated origin to all other destinations. Our preprocessing method is motivated by these observations.

For each origin, the preprocessing step proceeds as follows. We construct K different subproblems, where K is the number of destinations. In the k^{th} subproblem, $k = 1,..,K$, we partition the set of destinations into two groups: (i) a single destination, k, and (ii) the remaining $K - 1$ destinations which we aggregate into a "super-destination." Demands are aggregated across destinations within the super-destination, making appropriate adjustments for differences in travel times by modifying due date requirements to accurately reflect the latest departure date possible while ensuring on-time delivery for a given origin-destination pair. Weighted average fixed and variable costs are assessed for the direct and indirect routes between the origin and the aggregated destination.

This problem is now treated as an origin scheduling subproblem with two destinations. Direct and indirect train schedules for both the single (k^{th}) and the aggregated destination are derived, along with the corresponding container routing schemes, but only the direct train schedule for the k^{th} destination is retained. Therefore, at the end of this preprocessing step, for each origin, we have established direct train schedules for all K destinations. Having set the direct train variables for all origin-destination pairs in the preprocessing step, we solve the origin scheduling subproblems to determine indirect train schedules and all container flows.

This procedure generally will not provide an optimal solution to the original origin-scheduling problem because the direct train schedules are determined without full consideration of the details of the indirect train schedules. However, recall that the origin scheduling problem is an approximation in itself. From the viewpoint of solving the original problem, it would appear there is greater loss of optimality from decoupling the origins to create the origin scheduling subproblems (and from the inability of commercial software to find an optimal solution to the original origin scheduling subproblems) than there is from the use of this aggregation procedure in solving the individual origin scheduling subproblems.

5 Numerical Results

The purposes of our computational study are two-fold: to evaluate the performance of various decentralized methods on small problems for which optimal solutions can be found, and for more realistic problems, to compare the solutions of our various procedures with the best solutions obtained from commercial optimization software, and versus lower bounds.

The problems are solved on a Sun SparcStation 20 with 128 megabytes of RAM. We obtain solutions using all approaches described in Section 3 with CPLEX 6.0 as the underlying solver. For all executions of CPLEX, either for the monolithic problem or for the various subproblems arising in our decomposition procedures, we use a depth-first search, and strong branching, i.e., the choice of the branching variable is derived from solving a number of subproblems to determine which potential branch is likely to yield the greatest improvement in the objective function value. We also use the CPLEX built-in rounding heuristic, which attempts to determine an integer solution after every five nodes in the branch and bound tree. This combination of rules provided the best results for our set of problems. We also implemented a priority branching scheme, but it did not lead to significant performance improvement. We next describe the data used, and then report the results from these two studies.

Our main problem set consists of thirty instances with one hub, three to six origins and destinations, and with different container demand patterns and cost structures. We summarize problem characteristics in Tables 1 and 2. All problems have eight time periods in which containers become available at the origins. All trains have a capacity of 200 containers.

Container demand was generated for each origin-destination-arrival time-due date combination with a probability of 0.55 of being randomly generated from a discrete uniform distribution between 10 and 65, and a probability of 0.45 of being 0. Scenarios in which less expedited service was demanded were generated as described above, except that for each origin-destination-arrival time-due date combination such that the shipment necessitates transport via a direct train (i.e., $t + \alpha_{ik} \leq l < \beta_{ij} + \gamma_{jk} + \delta_j$), demands that were originally positive were independently set to zero with probability 0.4 to 0.5.

Industry data suggest that fixed and variable transportation costs for shipping a full train are approximately equal. We set the fixed charge associated with each train to be proportional to the distance, based on our observation that train operator labor constitutes the majority of this cost. Transportation costs are set appropriately based on the transit distance. Handling costs per container are based on an hourly wage of yard operators and the approximate time needed to load, unload, or rearrange a container. The yard storage cost per container per day is assigned a small value which provides incentive to ship earlier rather than later, all else being equal.

Because the monolithic versions of problems with the characteristics described above could not be solved optimally by CPLEX, we also generated a set of 30 "smaller" problems for our first computational study. These problems have three origins, one hub, and three destinations, and possess the cost structures given in Table 2. However, the container demand matrices are only about 12% as dense, i.e., roughly 20 to 25% as dense as in our main problem sets. Although these problems have fewer variables which will be set to non-zero values, the sparse demand matrices necessitate consideration of many different consolidation alternatives: consolidating demands across time for a single destination and sending them on a direct train, consolidating demands across destinations and sending them on a (usually) earlier indirect train, etc. When demands are either larger or the demand matrix is denser, fewer consolidation alternatives need to be considered, as train capacities are reached more quickly.

As shown in Table 3, in 12 of the 30 test cases (denoted problems 1-12 for convenience), CPLEX was not able to obtain an optimal solution to the monolithic version of the problem within a two and a half hour time limit. Thus, this set of "small" problems should not be regarded as easy. For problems 1-12 we report the ratio of the best integer solution found to the lower bound. The gaps average 5.6% but range up to 20%. Because we do not have optimal solutions for these problems, we do not compare our decomposition procedures against them, but use the other 18 problems for this purpose.

The other 18 problems generally could be solved to optimality relatively quickly (i.e., within a matter of seconds), although several problems require on the order of minutes to over an hour to solve. Note that even the problems requiring considerable CPU time for the centralized approach were not difficult to solve with our decomposition approaches. All variations of the decomposition procedures produce the same solution. For 16 of the 18 problems, our decomposition procedures produce optimal solutions. The two remaining problems deviate from the optimal by 3.8% and 8.2%, respectively. These deviations are due to the fact that the origin scheduling problem is both myopic and has "tunnel vision." That is, it neither sees the details of downstream costs outbound from the hub, nor can it anticipate consolidation opportunities that might be gained from coordinating shipments from other origins into the hub. Although gaps in the range of 4% to 8% are not trivial, they occur in only 10% of the problems. Moreover, because the problems are small, incorrect decisions about even a single train result in relatively large percentage increases in costs. Thus, these results should be regarded as providing very solid support for the performance of the heuristics. We now turn to the results for our main problem set.

We solved the 30 problems in our main problem set using the four decentralized approaches, the partially centralized approach and the centralized approach. As mentioned earlier, the centralized (monolithic) versions of the problems could not be solved optimally. We imposed a time limit of two and a half hours when solving the centralized versions. This time limit was based on the following observations: (i) the best identified integer solutions are often found in one to two hours of CPU time, (ii) the lower bounds do not improve substantially over their initial values, and (iii) even after several hours of CPU time, the gap between the best integer solution and the lower bound remains large, i.e., about 20%.

For the larger problems in this set (six origins and six destinations), the origin scheduling subproblems could not be solved optimally. To obtain a good solution, we used the heuristic preprocessing method described in Section 4. Recall that this procedure sets the direct train variables. Following this, remaining variables in the origin scheduling problem are determined optimally with the direct train variables fixed.

In Table 4, we report objective values for the first through fourth, and sixth approaches, expressed as a fraction of the objective value from the fifth approach, *partially centralized scheduling and routing*, which provided the best solution in all thirty realistic cases. We also report a lower bound, expressed as a fraction of the objective value from the fifth approach.

This lower bound is derived by including valid inequalities (cf. Newman and Yano, 1998) to strengthen the lower bounds provided by CPLEX. (Although these valid inequalities significantly improve the lower bounds, they contribute very little to improving CPU times, and require preprocessing effort to derive.)

The results show that in 80% of the problems, all decentralized approaches provide results at least as good as those obtained from the centralized approach with the straightforward application of CPLEX to the monolithic problem. The lower bound ratios clearly demonstrate that our heuristic procedures provide very good solutions in an absolute sense, i.e., no more than 10% from the optimum, and, on average, solutions within 6% of the optimal.

The container adjustment process for the *decentralized scheduling and routing* approach results in only modest improvements of well less than 1% on the average. In one case, the solution degraded. On the other hand, in many instances, the intermediate container adjustment led to improvements of roughly 1 to 2% in the results for *decentralized scheduling with ex post routing*. These results may suggest that the container routing decisions are secondary, and have a marginal effect on the solution once a train schedule has been established. The decision-maker would need to decide whether a fraction of a percent up to 2% improvement is worth the additional computational effort. However, for the six instances in which the centralized solution was superior to that obtained either from the decentralized approach or from the decentralized approach with ex post routing, for at least one of these two approaches, container adjustment yielded a solution commensurate in quality to that obtained from the centralized approach.

Table 5 contains summary CPU time statistics. We report a single run time statistic for the decentralized approaches because their CPU times are very similar. On the average, the decentralized approaches (approaches one through four) and the partially centralized approach (approach five) require a matter of seconds, or, at most, minutes, of CPU time to solve the smaller problems containing three to four origins and destinations. In contrast, the centralized approach requires over an hour, on average, to find the best solution attainable within a two and a half hour time limit. For problems with six origins and six destinations for which the preprocessing step is employed, the average computing time for the decentralized and partially centralized approaches increases to about 15 minutes. For the centralized approach, on the other hand, over one hour of CPU time, on average, was required to identify the solutions reported in Table 4, and the time limit of 9000 seconds was reached prior to confirming an optimal solution in all cases.

Thus, our procedures yield better solutions, on average, in significantly less time than the centralized approach. These decomposition procedures permit the evaluation of multiple scenarios to determine and quickly update weekly schedules in practice. Finally, our procedures allow for decision-making on a partially to completely decentralized basis, providing more flexibility for operational planners.

6 Conclusions

Although tradeoffs between centralized and decentralized decision-making are well understood in a qualitative sense, the cost tradeoffs have been studied in detail in only a few operational settings. We use as our framework a train scheduling and container routing problem that arises in rail intermodal operations. We present a spectrum of decision-making strategies, from completely decentralized (but with some consideration of downstream effects) to centralized. All of these approaches rely on the optimization of subproblems that vary with the degree of centralization. Our numerical results suggest that decentralized approaches may perform well.

For our application in particular, very good results can be obtained even with the use of the most decentralized approach. In this approach, each origin determines its schedule independently, using only a crude approximation of the costs incurred downstream for trains and containers sent via the hub. Then, train schedules and container assignments outbound from the hub are determined using the resultant container arrivals at the hub. Such a method would be relatively easy to implement. We demonstrate elsewhere (Newman and Yano, 1998) that even the most decentralized approach can provide for substantial cost savings (approximately 12%) over approaches with a similar degree of decentralization in which train schedules and container assignments are not optimized, but determined in a more *ad hoc* manner.

Acknowledgement

This work has been supported in part by National Science Foundation Grant GER/HRD 93-96288 to the University of California, Berkeley, and by funding from the United States Department of Transportation and the California Department of Transportation, awarded by the University of California Transportation Center. We also appreciate comments of the referees and Editor on an earlier version of this paper.

References

Aggarwal, A. K., M. Oblak and R. R. Vermuganti. "A Heuristic Solution Procedure for Multicommodity Integer Flows." *Computers and Operations Research*. Vol. 22, No. 10, 1995. pp. 1075-1087.

Anandalingam, G., K. Chatterjee and J.S. Gangolly. "Information, Incentives and Decentralized Decision-Making in a Bayesian Framework." *Journal of the Operational Research Society*. Vol. 38, No. 1, 1987. pp. 499-508.

Barnhart, Cynthia, Christopher A. Hane and Pamela H. Vance. "Integer Multicommodity Flow Problems." *Integer Programming and Combinatorial Optimization. 5th International IPCO Conference Proceedings.* Springer-Verlag: Berlin, Germany, 1996. pp. 58-71.

Barnhart, Cynthia and H. Donald Ratliff. "Modeling Intermodal Routing." *Journal of Business Logistics*. Vol. 14, No. 1, 1993. pp. 205-223.

Chu, S.C.K., C.K.Y. Lin and K.Y.K Ng. "Centralized versus Decentralized Manpower Resource Planning: The Case of a Hong Kong Company." *Journal of the Operational Research Society*. Vol. 42, No. 7, 1991. pp. 525-536.

Crainic, Teodor G. and Jean-Marc Rousseau. "Multicommodity, Multimode Freight Transportation: A General Modeling and Algorithmic Framework for the Service Network Design Problem." *Transportation Research*. Vol. 20B, No. 3, 1986. pp. 225-242.

Folk, Joseph F. and Amar K. Bharadwaj. "Conrail's Use of Linear Programming to Distribute Empty Freight Cars." *Proceedings of the Transportation Research Forum.* Vol. 21, No. 1, 1980. pp. 119-125.

Gazis, D.C. "Resource Allocation in a Large Decentralized Enterprise." *European Journal of Operational Research*. Vol. 30, No. 3, 1987. pp. 339-343.

Glickman, Theodore S. and Hanif D. Sherali. "Large-Scale Network Distribution of Pooled Empty Freight Cars over Time, with Limited Substitution and Equitable Benefits." *Transportation Research*. Vol. 19B, No. 2, 1985. pp. 85-94.

Gorman, Michael F. "An Application of Genetic and Tabu Searches to the Freight Railroad Operating Plan Problem." *Annals of Operations Research*. Vol. 78, 1998. pp. 51-69.

Gorman, Michael F. "Operating Plan Model Improves Service Design at Santa Fe Railroad." *Interfaces*. Vol. 28, No. 4, 1998. pp. 1-12.

Haghani, A. E. "Formulation and Solution of a Combined Train Routing and Makeup, and Empty Car Distribution Model." *Transportation Research*. Vol. 23B, No. 6, 1989. pp. 433-452.

Herrmann, J.W., G. Ioannou, J. M. Proth and I. Minis. "A Dual Ascent Approach to the Fixed-Charge Capacitated Network Design Problem." *European Journal of Operational Research*. Vol. 95, No. 3, 1996. pp. 476-490.

Hochbaum, Dorit S. and Arie Segev. "Analysis of a Flow Problem with Fixed Charges." *Networks*. Vol. 19, No. 3, 1989. pp. 291-312.

Keaton, M. "Designing Optimal Railroad Operating Plans: Lagrangian Relaxation and Heuristic Approaches." *Transportation Research*. Vol. 23B, No. 6, 1989. pp. 415-431.

Keaton, M. "Designing Railroad Operating Plans: A Dual Adjustment Method for Implementing Lagrangian Relaxation." *Transportation Science*. Vol. 26, No. 4, 1992. pp. 263-279.

Kendall, Kenneth E. and Barbara A. Schuldt. "Decentralizing Decision Support Systems." *Decision Support Systems*. Vol. 9, No. 3, 1993. pp. 259-268.

Kikuchi, Shinya. "Empty Freight Car Dispatching Model under Freight Car Pool Concept." *Transportation Research*. Vol. 19B, No. 3, 1985. pp. 169-185.

Kouvelis, P. and G. Gutierrez, "The Newsvendor Problem in A Global Market: Optimized Centralized and Decentralized Control Policies for a Two-Market Stochastic Inventory System." *Management Science*. Vol. 45, No. 5, 1997. pp. 571-585.

Kung, Shung-Kuang and James R. Marsden. "A Methodology and Experimental Shell for Formally Addressing Centralized/Distributed Decision Making Choices." *Decision Support Systems*. Vol. 15, No. 1, 1995. pp. 45-61.

Lamar, Bruce W., Yosef Sheffi and Warren B. Powell. "A Capacity Improvement Lower Bound for Fixed Charge Network Design Problems." *Operations Research.* Vol. 38, No. 4, 1990. pp. 704-710.

Leu, Y-Y., T.R. Rakes, L.P. Rees and W.A. Ceccucci. "Modeling Resource Allocation in a Decentralized Organization with an AI-Based, Goal-Directive Model." *Decision Sciences.* Vol. 23, No. 5, 1992. pp. 1027-1049.

Malone, Thomas W. "Modeling Coordination in Organizations and Markets." *Management Science.* Vol. 33, No. 10, 1987. pp. 1317-1332.

Malone, Thomas W. and Stephen A. Smith. "Modeling the Performance of Organizational Structures." *Operations Research.* Vol. 36, No. 3, 1988. pp. 421-436.

Marín, Angel and Javier Salmerón. "Tactical Design of Freight Networks. Part I: Exact and Heuristic Methods." *European Journal of Operational Research.* Vol. 90, No. 1, 1996. pp. 26-44.

Morlok, E.K. and R.B. Peterson. "Final Report on a Development of a Geographic Transportation Network Generation and Evaluation Model." *Journal of the Transportation Research Forum.* Vol. 11, 1970. pp. 71-105.

Newman, A. M. and Candace A. Yano. "Scheduling Direct and Indirect Trains and Containers in an Intermodal Setting." *Working Paper*, Department of Industrial Engineering and Operations Research, University of California, Berkeley, California, September 1998.

Nozick, Linda K. and Edward K. Morlok. "A Model for Medium-Term Operations Planning in an Intermodal Rail-Truck Service." *Transportation Research.* Vol. 31A, No. 2, 1997. pp. 91-107.

Obel, Børge. "On Organizational Design: From a Linear Programming Point of View." *The Journal of Management Studies.* Vol. 15, No. 2, 1978. pp. 123-137.

Palekar, Udatta S., Mark H. Karwan and Stanley Zionts. "A Branch-and-Bound Method for the Fixed Charge Transportation Problem." *Management Science.* Vol. 36, No. 9, 1990. pp. 1092-1105.

Schaffer, Joanne R. and Daniel E. O'Leary. "Use of Penalties in a Branch and Bound Procedure for the Fixed Charge Transportation Problem." *European Journal of Operational Research.* Vol. 43, No. 3, 1989. pp. 305-312.

Siebell, Stewart. "Centralized or Decentralized CIM?" *Computer-Aided Engineering.* Vol. 9, No. 10, 1990. p. 115.

Yano, Candace A. and A. M. Newman. "Scheduling Trains and Containers with Dynamic Arrivals and Due Dates." *Working Paper*, Department of Industrial Engineering and Operations Research, University of California, Berkeley, California, June 1998.

A Appendix: Subproblem Formulations

An origin scheduling subproblem for the single origin, ι, is formulated as follows:

$$(P_o^\iota) : min \sum_{ktl} h * I_{\iota ktl}^{o} + \sum_{ktl} c_{\iota k}^{a} * x_{\iota ktl}^{ao} + \sum_{jktl} c_{\iota jk}^{e} * x_{\iota jktl}^{eo} +$$
$$\sum_{ktl} g_\iota^o * x_{\iota ktl}^{ao} + \sum_{jktl} \widetilde{g_{\iota j}^{o}} * x_{\iota jktl}^{eo} + \sum_{kt} S_{\iota k}^{ao} * z_{\iota kt}^{ao} + \sum_{jt} \widetilde{S_{\iota j}^{eo}} * z_{\iota jt}^{eo}$$

subject to

$$b_{\iota ktl} + I_{\iota k(t-1)l}^{o} = I_{\iota ktl}^{o} + x_{\iota ktl}^{ao} + \sum_{j} x_{\iota jktl}^{eo} \qquad \forall k, t, l$$

$$\sum_{l} x_{\iota ktl}^{ao} \leq C * z_{\iota kt}^{ao} \quad \forall k, t$$

$$\sum_{kl} x_{\iota jktl}^{eo} \leq C * z_{\iota jt}^{eo} \quad \forall j, t$$

All variables restricted to be nonnegative and integer.

The hub scheduling problem for the single hub, ρ, and destination, κ, is formulated as follows:

$$(P_h^{\rho\kappa}) : min \sum_{itl} h * I_{i\rho\kappa tl}^{h} + \sum_{itl} g_\rho^h * x_{i\rho\kappa tl}^{h} + \sum_{itl} c_{i\rho\kappa}^{e} * x_{i\rho\kappa tl}^{h} + \sum_{t} S_{\rho\kappa}^{h} * z_{\rho\kappa t}^{h}$$

subject to

$$I_{i\rho\kappa(t-1)l}^{h} + x_{i\rho\kappa(t-\beta_{i\rho}-\delta_\rho)l}^{eo} = I_{i\rho\kappa tl}^{h} + x_{i\rho\kappa tl}^{h} \qquad \forall i, t \ni t \geq 1 + \beta_{i\rho} + \delta_\rho, l$$

$$\sum_{il} x_{i\rho\kappa tl}^{h} \leq C * z_{\rho\kappa t}^{h} \quad \forall t$$

All variables restricted to be nonnegative and integer.

Problem $P_h^{\rho\kappa}$ is solved using the solution $x_{i\rho\kappa tl}^{eo}$ from $\{\bigcup_i (P_o^i)\}$ as input data (demand).

B Appendix: Proof Regarding Effect of EDD constraints in Intermediate Container Adjustment

The intermediate container adjustment described in Section 3 entails adding constraints to the origin scheduling problem to ensure that (i) containers with a given due date are not shipped on a direct train unless all available containers with an earlier due date have been either shipped or already allocated to the direct train under consideration, and (ii) for each origin-destination pair, containers with a given due date are not shipped on an indirect train unless all containers with an earlier due date could be shipped, either on the current train, or on the next available direct train such that they arrive at the destination at least as early as if they had been allocated to the current train.

Below, we show that the imposition of these constraints serves to identify an alternate optimum with the EDD property, if an alternate cost-minimizing solution exists.

Part 1: Consider constraints of type (i) for direct trains from an arbitrary origin i to an arbitrary destination k in some time period t. Suppose that we have a cost-optimal schedule that does not satisfy constraints of type (i). Then there exists at least one container with a due date, say l which is assigned to (one of the) direct train(s) scheduled at time t, and another container with due date $l' < l$ which is assigned to an indirect train in t or later, or a direct train in $t + 1$ or later.

Now consider switching the assignments of these two containers. The assignments are feasible and the cost remains the same. Repeat this process until the EDD rule is satisfied for all containers. We have now constructed an alternate schedule with the same cost. Thus, the imposition of constraints of type (i) does not increase costs, if an alternate cost-minimizing solution exists.

Part 2: Consider constraints of type (ii) for an indirect train from an arbitrary origin i to the hub, and consider containers bound for an arbitrary destination, k. Suppose that we have a cost-optimal schedule that does not satisfy constraints of type (ii). Then there exists at least one container with a due date, say l, which is assigned to (one of the) indirect train(s) scheduled at time t, and another container with due date $l' < l$ which is assigned to an indirect or a direct train in $t + 1$ or later.

Now consider switching the assignments of these two containers. The assignments are feasible and the cost remains the same. Repeat this process until the EDD rule is satisfied for all containers. We have now constructed an alternate schedule with the same cost. Thus, the imposition of constraints of type (ii) does not increase costs, if an alternate cost-minimizing solution exists.

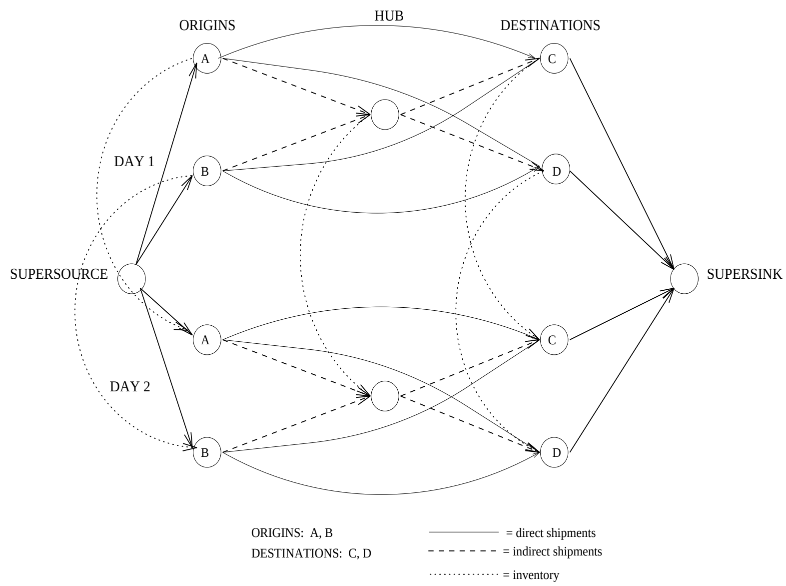

Figure 1: Multi-Commodity Network Depiction of our Problem

Table 1: Test Problem Characteristics

problems	number of origins-hubs destinations	relative proportion expedited service demanded
1-5	3-1-3	$\sim$ 20%
6-10	3-1-3	$\sim$ 10%
11-13	3-1-4	$\sim$ 20%
14-15	3-1-4	$\sim$ 10%
16-18	4-1-3	$\sim$ 20%
19-20	4-1-3	$\sim$ 10%
21-25	6-1-6	$\sim$ 20%
26-30	6-1-6	$\sim$ 10%

Table 2: Parameters for Test Problem Instances

parameter	range used in test problems
container arrival rate per day	0-65
fixed cost at origin (direct train) ($/train)	11000-15000
fixed cost at origin (indirect train) ($/train)	5000-8500
fixed cost at hub ($/train)	6200-9800
transportation cost ($/container)	40-100
handling cost ($/container)	1-2
inventory holding cost ($/container/day)	1.5-2

Table 3: Results for Small Problems

Problem	Centralized Objective / Lower Bound	Decentralized Objective / Optimum	Centralized CPU Time (sec.)
1	1.11	n/a	‡
2	1.07	n/a	‡
3	1.20	n/a	‡
4	1.02	n/a	‡
5	1.05	n/a	‡
6	1.06	n/a	‡
7	1.01	n/a	‡
8	1.01	n/a	‡
9	1.05	n/a	‡
10	1.03	n/a	‡
11	1.03	n/a	‡
12	1.03	n/a	‡
13	*	*	†
14	*	*	†
15	*	*	3974
16	*	*	†
17	*	*	†
18	*	*	†
19	*	1.04	†
20	*	*	†
21	*	*	†
22	*	*	†
23	*	*	3000
24	*	*	474
25	*	1.08	†
26	*	*	1023
27	*	*	†
28	*	*	†
29	*	*	†
30	*	*	†

* Indicates problem was solved to optimality
† Indicates CPU time is less than five seconds
‡ Time limit of 9000 seconds or memory limit is reached

Table 4: Ratio of Objective Value to that from the Partially Centralized Approach

Problem	Decentralized Scheduling and Routing	Decentralized Scheduling and Routing with Container Adjustment	Decentralized with ex post Routing	Decentralized with ex post Routing and Container Adjustment	Centralized	Lower Bound
1	1.00	†	1.00	†	1.02 ‡	0.98
2	1.00	†	1.00	†	1.00 ‡	0.98
3	1.00	†	1.00	†	1.01 ‡	0.96
4	1.00	†	1.00	†	1.02 ‡	0.99
5	1.00	†	1.00	†	1.00 ‡	0.97
6	1.00	†	1.00	†	1.01 ‡	0.93
7	1.00	†	1.00	†	1.00 ‡	0.90
8	1.01	1.02	1.01	1.01	1.05 ‡	0.93
9	1.00	†	1.00	†	1.03 ‡	0.92
10	1.00	†	1.00	†	1.03 ‡	0.94
11	1.00	†	1.00	†	1.01 ‡	0.94
12	1.00	†	1.00	†	1.01 ‡	0.97
13	1.00	†	1.00	†	1.01 ‡	0.96
14	1.00	†	1.00	†	1.04 ‡	0.91
15	1.02	1.02	1.02	1.02	1.02 ‡	0.91
16	1.01	1.01	1.01	1.00	1.00 ‡	0.97
17	1.00	†	1.00	†	1.01 ‡	0.97
18	1.00	†	1.00	†	1.00 ‡	0.96
19	1.00	†	1.00	†	1.03 ‡	0.93
20	1.00	†	1.00	†	1.02 ‡	0.93
21	1.02*	1.01*	1.02*	1.00*	1.01 ‡	0.93
22	1.02*	1.02*	1.02*	1.01*	1.01 ‡	0.94
23	1.01*	1.01*	1.01*	1.01*	1.01 ‡	0.94
24	1.01*	1.01*	1.01*	1.01*	1.01 ‡	0.95
25	1.01*	1.01*	1.01*	1.00*	1.01 ‡	0.95
26	1.02*	1.02*	1.02*	1.00*	1.01 ‡	0.92
27	1.02*	1.01*	1.02*	1.01*	1.02 ‡	0.91
28	1.02*	1.02*	1.02*	1.01*	1.01 ‡	0.91
29	1.03*	1.02*	1.03*	1.01*	1.00 ‡	0.90
30	1.01*	1.01*	1.01*	1.01*	1.01 ‡	0.91

* Indicates the preprocessing heuristic is used

† Indicates no benefit can be gained from container adjustment

‡ Terminated at a time limit of 9000 seconds

Table 5: CPU Times for the Decentralized Approaches, the Partially Centralized Approach and the Centralized Approach

Problem	Number of origins-hubs destinations	Decentralized Approaches (sec.)	Partially Centralized Approach (sec.)	Centralized Approach (sec.)*
1	3-1-3	†	†	1825 ‡
2	3-1-3	†	†	9000 ‡
3	3-1-3	†	†	6802 ‡
4	3-1-3	†	†	6051 ‡
5	3-1-3	†	†	4605 ‡
6	3-1-3	†	†	7892 ‡
7	3-1-3	142	115	8622 ‡
8	3-1-3	103	100	5662 ‡
9	3-1-3	†	†	192 ‡
10	3-1-3	†	†	5008 ‡
11	3-1-4	250	250	3824 ‡
12	3-1-4	97	97	1266 ‡
13	3-1-4	†	†	534 ‡
14	3-1-4	517	517	2258 ‡
15	3-1-4	2112	2112	4700 ‡
16	4-1-3	†	†	3008 ‡
17	4-1-3	†	†	683 ‡
18	4-1-3	†	†	497 ‡
19	4-1-3	†	†	2910 ‡
20	4-1-3	†	†	2800 ‡
21	6-1-6	972	951	1295 ‡
22	6-1-6	158	164	6469 ‡
23	6-1-6	114	120	1223 ‡
24	6-1-6	736	742	5870 ‡
25	6-1-6	810	818	1185 ‡
26	6-1-6	2492	2482	2642 ‡
27	6-1-6	1019	996	6383 ‡
28	6-1-6	912	886	1000 ‡
29	6-1-6	536	532	5498 ‡
30	6-1-6	670	680	8708 ‡

† Indicates CPU time is less than five seconds

* Time when best integer solution is first identified

‡ Time limit of 9000 seconds is reached without confirming optimality