Scheduling Tank Container Movements for Chemical Logistics

I. A. Karimi
Dept. of Chemical and Biomolecular Engineering, National University of Singapore, Singapore 117576

M. Sharafali
Dept. of Management, University of Melbourne, Parkville 3010, Australia

H. Mahalingam
The Logistics Institute—Asia Pacific (TLI-AP), National University of Singapore, Singapore 119260

DOI 10.1002/aic.10295
Published online in Wiley InterScience (www.interscience.wiley.com).

Effective logistics is the key to improving supply chain performance. Tank containers are attractive from the viewpoints of safety, cost, and environment, and are widely used for transporting fluid chemicals. Minimizing the logistics costs arising from the container flow imbalances across the globe and container cleaning is a major issue that chemical companies and affiliated third-party logistics firms face routinely. In contrast to dry containers, this important problem of managing tank containers in global chemical logistics has received no attention. An innovative, event-based, "pull" approach is presented for the minimum-cost or the maximum-profit scheduling of the transport and cleaning of multiproduct tank containers (loaded and empty) given a set of projected shipment orders. A novel linear programming formulation is developed and illustrated through examples that successfully solves large, industrially relevant problems with key practical considerations such as alternate ship schedules, delivery time windows, and intermodal transport routes. © 2004 American Institute of Chemical Engineers *AIChE J,* 51: 178–197, 2005
Keywords: tank container, repositioning, chemical logistics, intermodal transport, scheduling, shipping

Introduction

Logistics is the glue that binds the entities of a supply chain. An increasing number of businesses have now realized that its effectiveness is critical to the superior performance of the supply chain. For instance, the move toward just-in-time and agile manufacturing has tremendously increased the impact of the logistics decisions. This is especially true for the chemical industry with its globally distributed plant sites, storage terminals, suppliers, customers, and so forth. The global transport of chemicals and a variety of other materials (such as plant equipment, instrumentation, indirect materials, safety equipment, etc.) is central to its day-to-day operations. Logistics costs[1] can vary from 3.6% of the purchase price for a best-in-class (BIC) site to 20% at the other extreme.

The transport of chemicals by pipelines is ideal. However, this is not always possible. Chemical logistics requires a mix of means such as trucks, trains, ships, barges, tankers, and the like. Sea transport is the key to global chemical logistics. Bulk shipping of chemicals occurs in huge volumes, involving very large crude carriers (VLCCs) and a variety of multiparcel chemical tankers. The former routinely transport crude oils

Correspondence concerning this article should be addressed to I. A. Karimi at cheiak@nus.edu.sg.

worldwide,[2], whereas the latter transport clean petrochemical products (CPPs) and other chemicals such as vegetable oils.[3] However, there is another equally important segment of chemical logistics, called container shipping.

Containerization refers to the method of distributing materials in a unitized form, that is, a container that allows intermodal or multimodal transport (one that uses a mix of road, rail, sea, or inland waterways). The most common "dry" containers are 20 × 8.5 × 8 ft [designated as 20 feet equivalent unit (TEU)] and 40 × 8.5 × 8 ft [40 feet equivalent units (FEU)] boxes. These containers can be seamlessly loaded onto ships, trucks, or rail using appropriate handling machinery. Dry containers carry all kinds of goods, including certain solid chemicals, although liquid or gaseous chemicals require special containers, called tank containers. A tank container is simply a cylindrical tank set inside a frame of the standard dry container so that the machinery used for the dry containers can also handle the entire apparatus. This allows the tank containers to be stacked one atop another on ships, loaded on trucks, rail, and the like. A tank container offers several advantages over the conventional modes of shipping chemicals such as drums:

(1) It is environment-friendly because it minimizes the spillage during filling/unloading and leakage during transport. It permits the transport of dangerous chemicals in a safe manner.

(2) It is more cost-effective because it permits a higher payload compared to use of drums stowed in conventional dry containers (43% more volume). Its modular construction, ease of portability, and mechanized modes of handling all contribute to the cost savings.

(3) It allows multimodal transport.

(4) It is reliable, secure, and designed to last for 20–30 years.

(5) It is cleanable, reusable, and can be placed into alternate commodity service with minimum downtime.

(6) Customers who have limited space or wish to avoid the high cost of permanent storage can use it for temporary storage.

(7) It can be used for the transport of food-grade items.

Tank containers come in different sizes. Several organizations such as IMO (International Maritime Organization), ASME, and US DOT (U.S. Department of Transportation), to name but a few, have established manufacturing standards for these containers. Broadly, there are two major designs known as full-frame tank and beam-design tank. A standard tank container (TEU) can carry around 6300 gallons or 24 m^3. Another type of design, known as through-frame swap body tank that meets the ISO length requirements, is wider or higher, which allows higher volumes (27–35 m^3). The different types of tank containers (IMO 1, 2, 5, etc.) are classified according to the type of chemical being transported. Tank containers move a range of chemicals from food products such as palm oil, beer, and tomato concentrate, to liquefied gases such as refrigerants. To get an idea of the scale of the tank container operations, consider the fact that an existing big, multinational tank container company owns about 15,000 containers that make about 100,000 trips per year. The company considers this as a commodity business where margins are very thin, and even a slight improvement in the operations can make a substantial difference in the bottom line.

A major challenge that the companies using tank containers face arises from the imbalance of product supply and demand, and thus an imbalance in the container flows across different regions. There are major flows of loaded containers from the production centers toward the various demand centers globally. However, equivalent flows of products from the demand centers, which can enable the return of the emptied containers to the production centers, often do not exist. As a result, empty containers accumulate at the demand centers, which must be "repositioned" to the production centers. For instance, a company in the Asia–Pacific region ships loaded containers predominantly to western destinations, resulting in an accumulation of the empty containers in the United States and other European destinations. Unless the company brings back these empty containers to the Asia–Pacific region, it may not have enough empty containers for reuse. This incurs cost because ships do not carry empty containers free. The estimated cost of repositioning an empty container can vary from US$400 to US$800 per container. Some freight forwarders actually impose a surcharge (usually around US$300) for the empty container over and above the freight transport charge of the loaded one. The problem is further exacerbated by the need to clean the tank containers before reuse with the cleaning depots located away from the production centers, and the varied and stringent transport requirements of different chemicals. Similar imbalances of container flows exist even among different U.S. regions. Clearly, a systematic study and optimization of multiproduct tank container movements and related activities such as cleaning are crucial for reducing chemical logistics costs. Ensuring timely supplies of empty containers to production sites, shipping of loaded containers, cleaning and storage of emptied containers, and repositioning of empty containers to suitable places in anticipation of product orders must be done optimally to minimize the total logistics costs. This paper takes a comprehensive look at this important problem of tank container management.

In this paper, we address the container management problem from the perspectives of three companies:

(1) A chemical company that owns, or leases on a long-term basis, tank containers and manages them for its logistics needs.

(2) A third-party logistics (3PL) or fourth-party logistics (4PL) firm (such as Cendian) that either owns or leases containers, and manages them for its client chemical companies. Some chemical companies do not consider the packaging and transportation of chemical products as their core competencies, and are reluctant to invest in facilities and manpower. They prefer to outsource to 3PL and 4PL firms who manage their logistics activities.

(3) A container company (such as Stolt–Nielson) that owns containers, and undertakes the responsibility to deliver the chemical cargo as specified by its customers (chemical companies). In many cases, it also provides tank containers for chemical storage to various clients based on spot or contract requests.

We use the term *container operator* (or simply *operator*) to describe all of the above companies. Although our primary interest is in the tank containers used in shipping chemicals, the methodology presented here applies even to the containers used in shipping general dry cargos. This is why we use the terms *container* and *tank container* interchangeably.

A typical container movement starts with the receipt of an order by the operator from a chemical plant site (the origin site) for a required number of empty containers to deliver the chemicals by some delivery date at a destination site. The operator then has to decide from where to source the empty containers, when and how to deliver them to the origin site where they are loaded with the chemical cargo, when and how to transport the loaded containers to the destination site where they are unloaded, and where to clean and reposition the emptied containers. The repositioning may involve bringing the emptied containers back to where they started, diverting them to nearby areas that need them, or storing them at appropriate depots that can clean them and supply them to other sites in future.

Although the topic of dry container repositioning is not new, no earlier work (1) looks at the problem from the perspectives of the three companies described above, (2) addresses the management of tank ("wet") containers for global chemical logistics, (3) accommodates the variety of features as comprehensively as addressed in this paper, or (4) reports a continuous-time formulation. In addition to doing all these, this report presents a new linear programming methodology that is quite different from and is expected to be more efficient than the discrete-time approach that has been used so far in the literature for solving the deterministic problems of similar nature. Whereas the existing work uses a network flow modeling approach, we develop a novel, event-based "pull" approach that is more compact.

In what follows, we first critically review the existing literature on the container management problem. Next, we describe the problem in detail, and present a two-step procedure for developing a basic formulation with several simplifying assumptions. We use a small example to illustrate the basic model and the need for extensions. Then, we extend our basic formulation to relax most of the simplifying assumptions. Finally, we illustrate our approach through one industrial-scale example and one example with several extensions.

Literature Review

Most of the existing papers have focused on "dry" containers only. They have either considered the perspective of a shipping company or assumed a port-specific context. An early paper by White[4] used a network setting, and proposed an out-of-kilter algorithm that uses the properties of a dynamic transshipment network to optimize the flow of a single commodity through the network. White[4] showed that the network flow algorithm was relatively insensitive to the problem size when compared with linear programming (LP). Florez[5] extended the dynamic transshipment network approach to develop a model for empty container repositioning and leasing, and resolved it into two different LP problems. Crainic et al.[6] proposed a systematic modeling framework for these kinds of problems for a land-based distribution and transportation system. The framework included the space and time dependency of events and other characteristics such as substitutions, imports, exports, and so forth. Apart from a basic deterministic formulation, the authors proposed network models for dealing with uncertainty of demand and supply in the single commodity case. Shen and Khoong[7] developed a decision support system to solve a large-scale planning problem concerning the multiperiod distribution of empty containers for a shipping company. Besides container repositioning, the system also included cost-effective container leasing-in and off-leasing decisions. Abrache et al.[8] followed up on the work of Crainic et al.[6] by proposing a decomposition algorithm for the short-term planning of empty container movement and repositioning. Cheung and Chen[9] considered the dynamic empty container allocation (DCA) problem as a two-stage stochastic network, building up on previous work done by one of the authors on dynamic vehicle allocation (DVA). By exploiting the network structure, they demonstrated how a stochastic quasi-gradient method and a stochastic hybrid approximation procedure could solve the problem. It is noteworthy that the two principal references in this area focused only on land-based transportation[6] or sea-based (port-to-port) transportation.[9] Only Choong et al.[10] studied intermodal transportation networks in which they examined the effect of planning horizon on the mode of transport through an integer program developed for minimizing the cost of empty container repositioning.

In addition to the empty container positioning, Chen and Chen[11] studied the problem of loaded container movements, and Chen and Ma[12] studied its stochastic version. Lai et al.[13] resorted to simulation to analyze this problem and proposed some heuristic search techniques.

The empty-container repositioning problem is also relevant to manufacturers whose products require the use of returnable or recyclable containers. A soft drink company is a typical example. For works in this area, the interested reader can refer to Del Castillo and Cochran[14] and the references therein. The authors model and analyze the reusable bottle production and distribution activities of a large soft drink manufacturer. Such systems can have several operating policies (see Lutzebauer[15] and Kroon and Vrijens[16]). In a *switch pool system*, participants have their own allotments of containers and are responsible for upkeep and maintenance. Whenever a member of the pool receives a shipment, an equal number of empty containers must be shipped back to the supplier. If the carrier is also allocated customers, then the carrier replaces the loaded containers with empty containers when it picks up a load from a supplier or vendor. In *systems with return logistics*, a central agency owns the containers. It is the agency's responsibility to return the containers after they are unloaded. Usually, the agency consolidates empty containers at the customer locations before shipping the accumulated lots over time. Two variants of this system are in vogue. One is the *transfer system* and the other is the *depot system*. Under the transfer system, the sender always uses the same containers. It is the supplier's responsibility to track and trace the containers and also to maintain, clean, and store them. Under the depot system, the empty containers always go to a container depot, which also supplies the containers on demand. In the third system, called the *system without return logistics*, a central agency owns the containers, and the customers rent them for fixed periods. The containers return to the agency after use.

It is interesting to note that some of the above policies also exist in the cargo container industry. Furthermore, with the help of information technology, companies can now spot-lease containers from the market quite easily. The model considered in this paper assumes the operator using a system with return logistics with the additional option of spot leasing.

Problem Description

For the global container movements, we assume that the transportation network (Figure 1) consists of depots, seaports

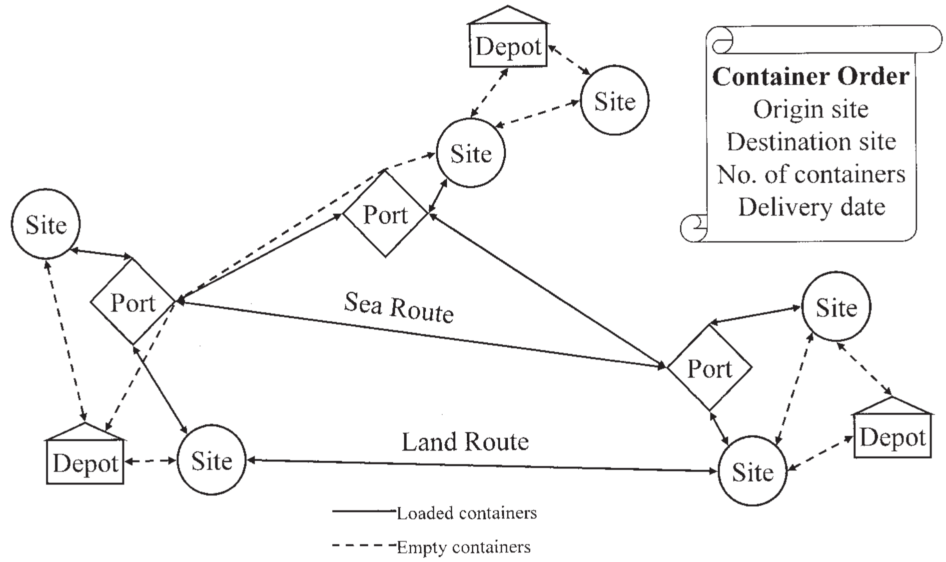

Figure 1. Problem overview and transportation network for the container movements.

(or simply ports), and sites. A site is a facility that receives empty (loaded) containers, and loads (unloads) them. It can be a manufacturing, supplier, 3PL, loading, retail, distribution, or customer facility. A depot is a facility that receives empty containers from other depots or sites, cleans and stores them, and then sends them to sites or other depots as and when needed. A freight forwarder usually picks the loaded (empty) containers from a site (depot) and ships them to another site (depot). A site may or may not hold empty containers for a limited time. A site-to-site container movement in general involves three transportation legs. Normally, a land-based leg takes the containers from an origin site to an origin seaport, then a sea-based leg takes them to a destination seaport, and finally another land-based leg takes them to a destination site. Of course, the same may also apply to a site-to-depot or depot-to-depot movement of an empty container.

Let D be the number of depots ($d = 1, 2, \ldots, D$) and S be the number of sites ($s = 1, 2, \ldots, S$). We assume that the operator knows the current state of the system at time zero. In other words, it knows the empty-container inventories at various depots and sites, and the containers (loaded or empty) that are in transit and their target destinations. At present, the operator has O known, consolidated orders ($o = 1, 2, \ldots, O$) for empty containers from various sites. Each order o has four attributes:

(1) Origin site (m): the site that has placed the order for empty containers and will load them after receiving

(2) Destination site (n): the site that will receive the loaded containers

(3) Order quantity (NC_o): the number of containers that site m has requested in order o.

(4) Due date (DD_o): the time at which the loaded containers must reach site n

By a consolidated order, we mean that multiple orders with identical m, n, and DD_o are lumped into a single bigger order. The operator (or the company that owns the containers) aims to devise a detailed schedule of container movements that minimize the total operating cost. This involves:

(1) Assigning the depots or sites from which to send empty containers to the origin site for each order, fixing the numbers of such containers and setting the times at which they should leave.

(2) Setting the number of containers that the operator should spot-lease for each order, if it is unable to fill the order with its own containers.

(3) Selecting the depots or sites that should receive the containers released by the destination site for each order, fixing the numbers of such containers and setting the times at which they should reach.

(4) Computing the minimum cost of container movements required to fill all the O orders.

We now make two sets of assumptions. Although the first set is permanent and central to this paper, the second is temporary merely to simplify the development of a basic model. Later, we extend the basic model by relaxing the temporary assumptions.

Assumptions

Set A: Primary or Permanent

[A1] The operator may spot-lease containers from the open market at some higher charge, in case it does not have its own containers to fill an order.

[A2] The open market has an unlimited supply of empty containers at all times.

[A3] The system is deterministic. The times and costs for moving the empty and loaded containers between depots/sites and sites are known a priori. Similarly, the costs for holding empty containers temporarily at sites are also known.

[A4] All ship schedules are known.

[A5] Container loading and unloading times are either negligible compared to their depot-to-site or site-to-site transport times or are included appropriately in their transit times based on the numbers of containers.

Set B: Secondary or Temporary

[B1] The company fills each order fully and just in time (JIT), that is, all shipments reach their destinations fully at their specified times.

[B2] No containers are in transit at time zero.

[B3] A site may hold empty containers temporarily at some cost. This would allow the site to reuse containers from its own inventory.

[B4] A site, after unloading some containers, need not always return them to the depots. It may send them directly to a demand (origin) site.

[B5] Explicit depot-to-depot container movements do not exist. However, this does not mean that a depot cannot supply empty containers to another depot that in turn may supply to a nearby demand site. In other words, a depot may supply a site through another depot, but the corresponding depot-to-depot movement is not explicit in our model.

[B6] Containers do not require cleaning.

[B7] Normally, several "ways" or "modes" exist for moving containers from one site/depot to another site. These may arise from the various combinations of origin, destination and transshipment ports, various modes of land transport, and so forth. These will typically incur different costs and require different times. We assume that we have one predetermined "best" route for every container move, and we know the time and cost (per container) of transportation by that route.

[B8] There is only one container type. All containers are indistinguishable; we can substitute one for another.

[B9] Ships can carry unlimited number of containers.

[B10] The holding costs are the same (or zero with no loss of generality) at all depots. Therefore, there is no explicit depot-to-depot container movement.

[B11] Leases are only spot leases. No short-term or long-term leases are considered.

Solution Methodology

As in most scheduling problems, time representation is the key. In this work, we use a novel event-based representation that treats time as continuous. Batch process scheduling literature has used continuous-time representation in several forms. These include variable-length slots with[17] and without[18] fixed events, and variable events without slots.[19] In contrast to the many scheduling problems in which it is impossible to fix the possible event times a priori, the present problem is unique in that it permits, with the aforementioned assumptions and simplifications, a continuous-time representation in which we can fix all possible event times a priori.

Our methodology constitutes two phases. In the first phase, we generate a chronologically ordered superlist of possible instances at which container (empty or loaded) movements (events) may occur. We also identify the types of movements that may occur at each such instance. In the second phase, we use these superlists of times and events to develop a linear programming (LP) formulation whose solution will pick the events that minimize the schedule cost. We begin with Phase 1. For the time being, we allow a container to reach any site or depot from any other site and vice versa.

Phase 1: events and event times

The most important feature of the present system is that it is order-driven; that is, the loaded containers specified in the orders must reach their respective destination sites on time. To attain this objective, we trigger container movements (or events, both backward and forward in time) at appropriate points (sites, ports, and depots) and times. For instance, empty containers departing a given depot and bound for a specific site is an event. However, because we do not know a priori the "optimal" events that may lead to the minimum operating cost, we identify a superlist of all possible, acceptable events that may occur in the future. Now, knowing the transport times and the times for various activities (customs clearance, cleaning, etc.) between the events, we fix the exact times at which these events must occur to make the entire system operate under the JIT policy. Note that the JIT policy is not necessarily optimal. In fact, we later see an example where the total cost reduces when we relax the JIT policy. We now let the optimizer pick the events that should constitute the optimal schedule. Let us illustrate this event generation process for an arbitrary order o.

Figure 2 describes all the events that may occur at different times in connection with an order o. Order o demands NC_o loaded containers to reach site n at time $t1 = DD_o$ from site m. We define the arrival of loaded containers at site n from site m as an event. We consider this as the primary (or the first) event that triggers all other events, both backward and forward in time. Because we assume JIT operation, this primary event occurs at $t1$. Thus, $t1$ is the time for this first event associated with order o. If the containers move by scheduled ships, then it may be impossible for them to arrive exactly at DD_o because the time may not match with the available ship schedules. In such a case, we take $t1$ as an acceptable time that matches with ship arrivals and at which the containers can actually arrive. Note that this may be earlier or later than DD_o.

For the loaded containers to reach site n, another event must happen. It is the arrival of empty containers at site m, the second event associated with order o. As stated earlier, we have predetermined the best transport route or "mode" between sites m and n. So let $TSS(t1, o, m, n)$ denote the total time required between the arrivals of the containers at sites m and n using this route for order o. Note that $TSS(t1, o, m, n)$ may be time-dependent and order-dependent. As illustrated in Figure 2, it includes the times for loading the containers at site m, moving them from m to a nearby origin port, export clearance at the origin port, transporting them from the origin port to a destination port, import clearance at the destination port, transport from the destination port to site n, and any other expected delays on the entire journey. It also depends on the schedules of the intermediate transporters and the ship that will carry the loaded containers. In brief, given that the loaded containers must reach site n at $t1$, we fix a time $t2 = t1 - TSS(t1, o, m, n)$ for the second event associated with order o.

Now, the second event necessitates several events that may occur before $t2$. Clearly, one or more empty-container arrivals must occur at site m and time $t2$, so logically their predecessor dispatch events must occur before $t2$. To identify these events, consider the four possible ways in which empty containers may reach site m at $t2$:

[e1] Site m may already be holding some empty containers. Assuming that it can use them instantaneously, we do not define a separate event for this supply.

[e2] Another site s may be holding empty containers that it can supply to site m. Therefore, the third set of possible events associated with an order o consists of the departures of empty containers from other sites to site m. Because the empty containers must reach (JIT) site m at $t2$, we can fix the time $t3(s) = t2 - TSSE(t2, o, s, m)$ at which the empty containers must leave site s to reach m at $t2$. Here $TSSE(t2, o, s, m)$ is the time required between the departure of empty containers from site s and their arrival at site m in connection with order o. We fix $t3(s)$ as the occurrence times for the third set of events associated with order o.

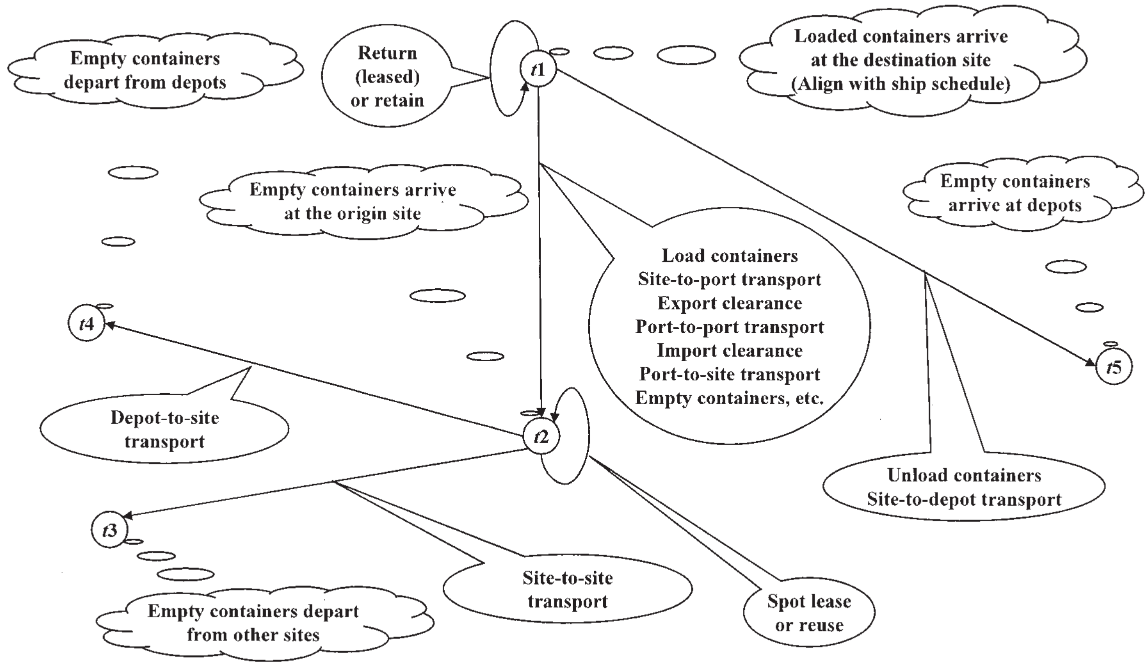

Figure 2. Generation of events and event times for an order.

[e3] Very likely, one or more depots may supply empty containers to site *m*. Therefore, the fourth set of possible events associated with an order *o* constitutes the departures of empty containers from depots to site *m*. These events occur at all depots that can possibly supply empty containers to site *m*. As discussed earlier, some depots may actually supply by some nearby depots. However, regardless of how the empty containers reach site *m*, we can always compute $t4(d) = t2 - TDS(t2, o, d, m)$ as the times for this fourth set of events. Here $TDS(t2, o, d, m)$ is the total time needed between the departure of empty containers from depot *d* and their arrival at site *m* in connection with order *o*.

[e4] Finally, the company may lease empty containers from the spot market. Because the spot market has an unlimited supply, we safely assume that the leased containers arrive exactly at *t*2 and do not define any new event for this. The spot-lease would dictate where (the destination site *n* or the origin site *m*) these containers must return to the leasing company.

Having generated the possible events before the arrival of order *o* at site *n*, we now identify the events that may occur afterward. On their arrival at *t*1, site *n* would first unload the containers. Now, there are three ways of dealing with these emptied containers.

[f1] Some containers may be on lease from the spot market and they must return to the leasing company. Therefore, we have an event that is the return of the leased containers to the leasing company from site *n*. Because the unloading time is negligible, this event occurs at *t*1. Now, the lease may dictate where these containers should return to the leasing company. If the operator can return them at site *n*, then it incurs no extra cost. However, if it must return them at site *m*, then it incurs the cost of transporting the empty containers back. However, we do not need an extra event for this return because only the cost matters and not the time.

[f2] The remaining emptied containers belong to the operator. Some of them may remain temporarily at site *n* for the use by some near-future orders. However, we do not need a separate event for this.

[f3] Very likely, most operator-owned emptied containers would return to one or more depots. Thus, we have the fifth set of events representing the arrivals of empty containers from site *n* to various depots. The time at which the emptied containers arrive at depot *e* from site *n* is $t5(e) = t1 + TSD(t1, o, n, e)$. $TSD(t1, o, n, e)$ is the total time between the arrivals of loaded containers at site *n* and emptied containers at depot *e*. Note that it depends on *t*1, and includes the times for unloading the containers at *n* and transporting them to depot *e* as shown in Figure 2.

We now see from Figure 2 that an order *o* may trigger at most $2D + S + 4$ events (container arrivals/departures at sites/depots) excluding those to or from the site inventories. These events may occur at a maximum of $2D + S + 2$ distinct times. Generating the events for all orders using the above procedure, we get at most $O(2D + S + 4)$ events and $O(2D + S + 2)$ event times. These events and their times form the basis for our LP formulation for this scheduling problem.

Phase 2: novel LP formulation

As a prerequisite to our formulation, we process the events and event times as follows. We discard all the events with negative event times, and their associated event times. We then sort all the event times in the chronological order and remove the duplicate entries. This gives us a strictly increasing superlist of event times at which the various events may occur. We use $t_0 (=0) < t_1 < \cdots < t_{k-1} < t_k < t_{k+1} < \cdots < t_K$ to denote this ordered superlist of event times. Next, we identify all the events that may possibly occur at each t_k ($k \geq 0$) in the list. For each such event, we assign a specific variable (*x*, *y*, *u*, *v*, and *w* as shown in Figure 3) to denote the number of containers involved in the event as follows:

- x_{dmjk} ($j \leq k$): the number of empty containers leaving depot *d* at t_j to reach site *m* at t_k

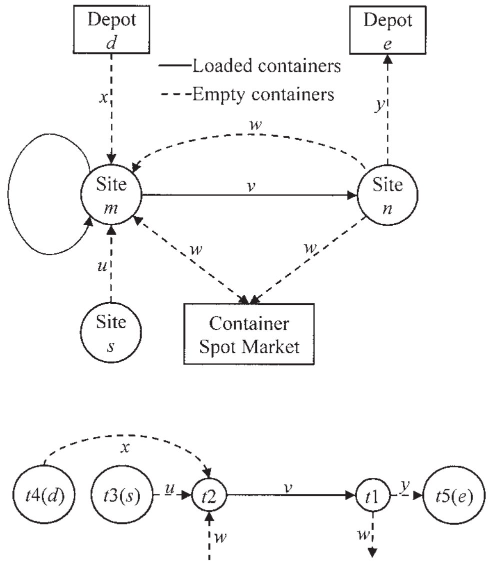

Figure 3. Container flow variables associated with a shipment delivery from site *m* to *n* at different times *t*1, *t*2, *t*3, *t*4, and *t*5.

- y_{nejk} ($j \le k$): the number of empty containers leaving site n at t_j to reach depot e at t_k
- u_{smjk} ($j \le k$): the number of empty containers leaving site s at t_j to reach site m at t_k
- v_{mnjk} ($j \le k$): the number of loaded containers leaving site m at t_j to reach site n at t_k
- w_{mnjk} ($j \le k$): the number of leased (from the spot market) containers in v_{mnjk}

In the above, x, y, and u involve the operator-owned containers only. Because each order o has unique sites m and n, a known container requirement NC_o, and known times $t1 = DD_o$ and $t2$, we get v_{mnjk} ($j \le k$) $= NC_o$ for ($t_j = t2$, $t_k = t1$) and zero otherwise. Thus, all v_{mnjk} ($j \le k$) are known and not variables in this basic formulation. Finally, we use DI_{dk} to denote the number of empty containers at time t_k in depot d and SI_{sk} to denote the same at site s.

Because the x-variables represent the empty containers leaving a depot d, and the y-variables represent those entering d, a container balance for a depot d at t_k yields

$$DI_{dk} = DI_{d(k-1)} + \sum_{s} \sum_{j \le k} y_{sdjk} - \sum_{s} \sum_{j \ge k} x_{dskj} \qquad (1)$$

with a capacity restriction, $DI_{dk} \le DI_d^U$.

In contrast to a depot, a site can also send/receive containers to/from the depots, sites, or spot market. Therefore, the following container balance at a site s involves all the variables:

$$SI_{sk} = SI_{s(k-1)} + \sum_{d} \sum_{j \le k} x_{dsjk} + \sum_{n \ne s} \sum_{j \le k} u_{nsjk} + \sum_{m \ne s} \sum_{j \le k} (v_{msjk} - w_{msjk}) - \sum_{d} \sum_{j \ge k} y_{sdkj} - \sum_{m \ne s} \sum_{j \ge k} u_{smkj} - \sum_{n \ne s} \sum_{j \ge k} (v_{snkj} - w_{snkj}) \qquad (2)$$

with a capacity restriction, $SI_{sk} \le SI_s^U$.

Because the number of containers spot-leased for a given order cannot exceed the total number of containers in that order, we have the upper bound, $w_{mnjk} \le v_{mnjk}$, $j \le k$. Equation 2 essentially ensures that the spot-leased containers never enter the inventory at any site. These containers simply appear at the origin site for loading and disappear from the destination site after unloading. Thus, they cannot remain in the system for future use.

Toward the end of the scheduling horizon, an emptied container may stay put at its destination site to avoid the cost of moving back to a depot. To prevent this behavior, we require that all containers eventually return to the depots. Therefore,

$$\sum_{d} DI_{d0} = \sum_{d} DI_{dK} \qquad (3)$$

Finally, the scheduling objective is to minimize the total cost of managing the containers, which is given by

$$C = \sum_{d} \sum_{s} \sum_{k} \sum_{j \ge k} x_{dskj} XC_{dskj} + \sum_{s} \sum_{d} \sum_{k} \sum_{j \ge k} y_{sdkj} YC_{sdkj} + \sum_{s} \sum_{n \ne s} \sum_{k} \sum_{j \ge k} u_{snkj} UC_{snkj} + \sum_{m} \sum_{n \ne m} \sum_{k} \sum_{j \ge k} v_{mnkj} VC_{mnkj} + \sum_{m} \sum_{n \ne m} \sum_{k} \sum_{j \ge k} w_{mnkj} WC_{mnkj} + \sum_{s} \sum_{k<K} h_s SI_{sk}(t_{k+1} - t_k) \qquad (4)$$

where WC_{mnkj} is the total cost (the lease cost plus the cost of transporting back if needed) for leasing one empty container from the spot market at site m and time t_k; XC_{dskj} is the cost of sending one empty container from depot d at time t_k to reach site s at time t_j; YC_{sdkj} is the cost of sending one empty container from site s at time t_k to reach depot d at time t_j; UC_{snkj} is the cost of sending one empty container from site s at time t_k to reach site n at time t_j; and h_s is the cost of holding one empty container at site s for a unit time. Note that the transportation costs for the empty and loaded containers are different. Even though the cost of moving the loaded containers is fixed, Eq. 4 includes it for the sake of consistency. In fact, as discussed later, it becomes essential to include that cost for some extensions.

Equations 1–4 constitute our LP formulation for the basic problem. Note that most variables included in the formulation are zero because only the variables that correspond to possible events exist. Thus, even though we have used sums over all possible time indexes, not all variables will exist in the formulation. As long as the data for container demands are integers, all variables will assume integer values in the optimal solution. Note that we did not assume any prespecified scheduling horizon. The last possible event fixes the scheduling horizon in our approach. In this way, the horizon allows all containers to return to the depots.

A glance at the constraints of our LP reveals that the coefficients of the variables are 1, 0, or −1, and thus the coefficient matrix is fully unimodular. Then, from the proof in Bazaara and Jarvis,[20] the integrality of an optimal solution to the above LP is guaranteed.

Before we extend our approach to relax our temporary assumptions, we consider a small example to illustrate the application of our model and the impact of the JIT assumption.

Table 1. Order Data for Example 1

Order o	Origin Site m	Destination Site n	Containers NC_o	Due Date DD_o (h)
1	7	4	24	141
2	6	10	16	205
3	2	9	17	297
4	7	10	22	181
5	5	8	8	167
6	9	8	5	46
7	8	3	9	63
8	2	4	10	208
9	3	9	19	75
10	4	9	27	119

Example 1

We have five depots ($D = 5$), 10 orders ($O = 10$), and 10 sites ($S = 10$). Table 1 lists the order data, Table 2 lists the various costs, Table 3 gives the transport times, and Table 4 shows the inventory levels at various depots. The sites have no containers at time zero and no containers are in transit.

To illustrate the event generation procedure, consider order 5. For this order, eight loaded containers must reach site 8 at 167 h from site 5. Therefore, $m = 5$, $n = 8$, $NC_5 = 8$, $DD_5 = 167$ h, and $v_{5,8,150,167} = 8$. The various event times are $t1 = 167$ h, $t2 = 150$, $t3(1) = 139$, $t3(2) = 131$, $t3(3) = 138$, $t3(4) = 142$, $t3(5) = 150$, $t3(6) = 129$, $t3(7) = 139$, $t3(8) = 133$, $t3(9) = 141$, $t3(10) = 137$, $t4(1) = 138$, $t4(2) = 131$, $t4(3) = 129$, $t4(4) = 137$, $t4(5) = 139$, $t5(1) = 183$, $t5(2) = 177$, $t5(3) = 198$, $t5(4) = 184$, and $t5(5) = 184$. Generating such

Table 3. Transport Times (h) for the Containers in Example 1

Site	Site 1	2	3	4	5	6	7	8	9	10
	Site-to-site transport time									
1	0	16	7	9	11	26	6	17	13	18
2	16	0	15	18	19	34	16	10	17	20
3	7	15	0	10	12	28	7	16	13	18
4	9	18	10	0	8	23	9	17	11	16
5	11	19	12	8	0	21	11	17	9	13
6	26	34	28	23	21	0	26	31	22	22
7	6	16	7	9	11	26	0	17	13	18
8	17	10	16	17	17	31	17	0	15	16
9	13	17	13	11	9	22	13	15	0	11
10	18	20	18	16	13	22	18	16	11	0
Depot										
	Depot-to-site transport time									
1	7	15	6	10	12	28	7	16	13	18
2	16	6	15	18	19	34	16	10	17	20
3	26	34	28	23	21	6	26	31	22	22
4	18	20	18	16	13	21	18	17	11	6
5	6	16	7	9	11	26	6	17	13	18

Table 2. Costs (US$) for Transporting, Holding, and Spot-Leasing the Containers in Example 1*

Depot	Site 1	2	3	4	5	6	7	8	9	10
	Depot-to-site transport cost per empty container									
1	107	428	428	225	313	931	107	447	359	555
2	481	50	427	549	574	1192	481	218	522	615
3	883	1193	931	756	684	684	883	1079	721	703
4	551	638	562	453	352	675	551	489	255	78
5	551	482	107	180	280	883	883	488	346	548
Site										
	Site-to-site transport cost per empty container									
1	0	482	107	180	280	883	883	488	346	548
2	482	0	428	550	575	1193	482	219	522	615
3	107	428	0	225	313	931	107	447	359	555
4	180	550	225	0	157	756	180	511	254	456
5	280	575	313	157	0	684	280	500	169	358
6	883	1193	931	756	684	0	883	1079	721	703
7	551	482	107	180	280	883	0	488	346	548
8	488	219	447	511	500	1079	488	0	416	464
9	346	522	359	254	169	721	346	416	0	253
10	548	615	555	456	358	703	548	464	253	0
Site										
	Site-to-site transport cost per loaded container									
1	0	503	110	187	291	923	923	509	360	572
2	503	0	446	574	600	1247	503	227	545	642
3	110	446	0	233	326	973	110	466	374	579
4	187	574	233	0	162	789	187	533	264	475
5	291	600	326	162	0	714	291	522	175	372
6	923	1247	973	789	714	0	923	1128	753	734
7	575	503	110	187	291	923	0	509	360	572
8	509	227	466	533	522	1128	509	0	434	484
9	360	545	374	264	175	753	360	434	0	263
10	572	642	579	475	372	734	572	484	263	0

*Holding cost per container at all sites = $10/h; spot-lease cost = $1000 per container.

Table 4. Initial and Final Inventory Levels at the Depots in Example 1

	Inventory	
Depot	Initial	Final
1	5	0
2	6	0
3	13	0
4	10	37
5	3	0

events and times for all 10 orders, we find that the last event could occur at 319 h. Thus, we have a scheduling horizon of 319 h.

We modeled the LP for this example in GAMS 20.7.133[21] and solved it using CPLEX 7.5 on a Dell workstation with dual Intel Xeon 2.8-GHz processors and 2 GB RAM. The model involved 2051 continuous variables, 1817 constraints, and 5486 nonzeros. Its solution required 81 LP iterations and 0.046 s of CPU time. The minimum total cost for all container movements is $143,652.

Figure 4 shows how the optimizer meets the empty container demands of various orders. Of the total demand of 157 empty containers, the depots supply 75 containers, whereas 11 are leased (order 2) from the spot market. Although the depots meet nearly 50% of the demand, we find that it is often cheaper to temporarily hold emptied containers at the destination sites, and then move them directly to demand sites rather than send them by depots. This happens in spite of the nonzero holding costs at sites. In fact, sites supply 71 of the total 157 containers in this example. For instance, 27 loaded containers of order 10 reach site 9 at 119 h. After unloading them, the optimizer moves 24 containers to site 7 to meet the demand for order 1. At other times, a site holds some containers for its own near-future use. For instance, five loaded containers of order 6 reach site 8 at 46 h from site 9. Site 8 holds them until 47 h, and uses them for order 7 that needs nine empty containers. Note that the reason why the sites supply or reuse containers in this example is that container cleaning is absent. If cleaning is mandatory after every loaded movement, and if the sites cannot clean the containers, then all containers must return to the depots and no site can supply containers even to itself.

Table 4 lists the initial and final inventory levels at the depots. Interestingly, all the 37 containers end up at depot 4. This is because sites 9 and 10 are the last to receive the loaded containers and depot 4 happens to be the least cost option for both to return the containers to the depots as required by Eq. 4.

In the above solution, we assumed that the orders arrived exactly at the due dates. Now suppose that the orders allowed some flexibility in delivery dates, that is, time windows for delivery instead of a single delivery date. If we assume that all orders in this example allowed a delivery time window of 10 h, then how will the minimum cost change? For instance, assume that order 1 could be delivered any time within [140, 150] instead of 141, order 2 within [200, 205], and so on. In this scenario, we find that the total cost reduces to $136,663, that is, a saving of nearly $7,000. This shows that the solution based on the JIT assumption may not be optimal, when the customers allow flexible deliveries.

The reason for the above cost reduction is as follows. Although the containers arrive just in time (JIT) at a destination site to avoid holding, they may arrive too late for some nearby depots or sites that need empty containers. In this case, the operator has to borrow empty containers from the spot market, incurring a higher cost. The operator can avoid this additional

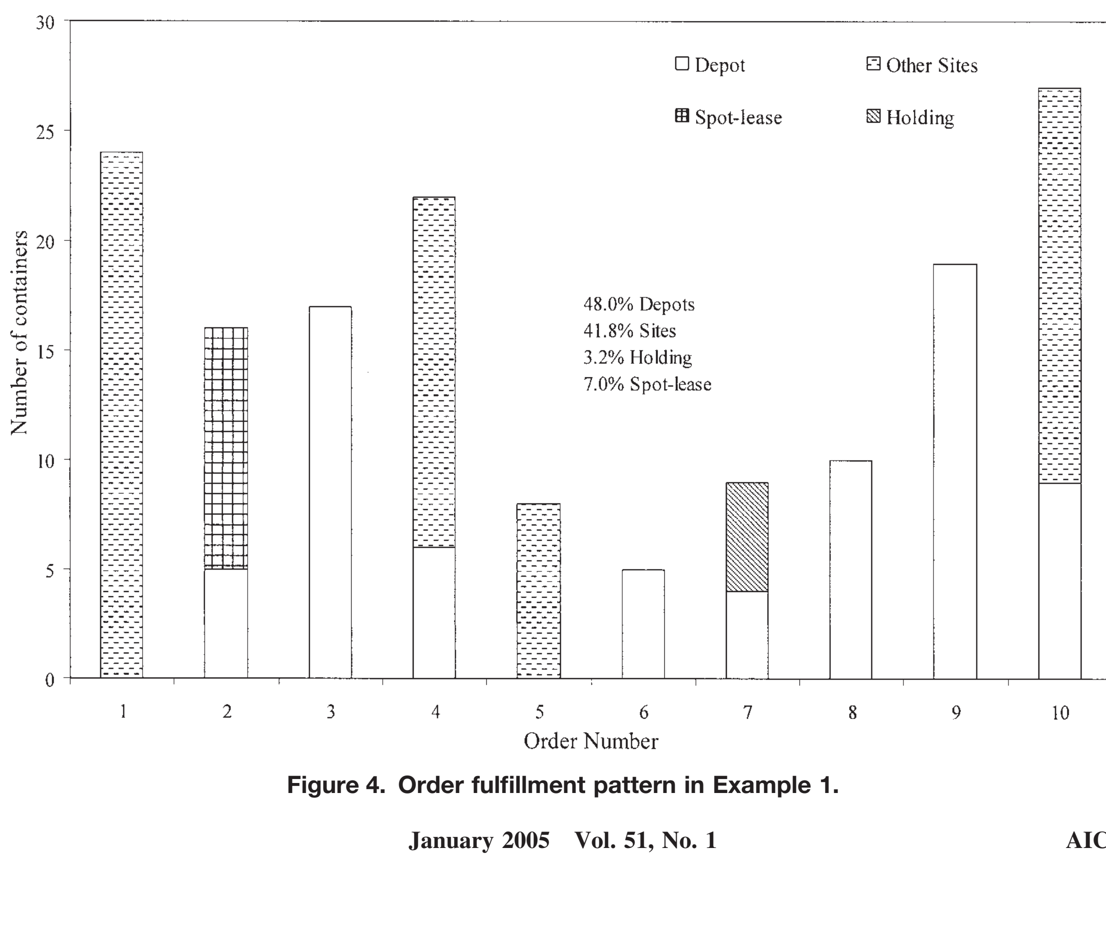

Figure 4. Order fulfillment pattern in Example 1.

cost, if the destination site is flexible with its delivery date, making it possible for the containers to arrive a bit early. It is in this context that delivery time windows become important. Furthermore, order due dates are often time windows rather than specific dates. For instance, a receiving site may demand that the loaded containers arrive within a time window of say 3 days. Thus, if the customers are flexible with their delivery dates, then they normally specify time windows within which they expect order deliveries.

We note that even with the inclusion of flexible delivery dates as above, the formulation remains an LP, albeit the length of the superlist increases and LP is bigger. The same will be true with time windows for both destinations and sites. We now extend our model to handle this and other relaxations of the temporary assumptions (Group B) listed earlier.

Extensions

We begin with the reality of multimodal transportation.

Multimodal transportation

Most container movements involve a combination of land, sea, or air transports. Note that our model did not explicitly assume any mode of transport. Therefore, it applies seamlessly to any multimodal transport route. We need some modification only to handle ship schedules and capacities, which we discuss later. In general, the only data that our model requires are the time and cost required for the transport, which we can compute for any multimodal transport route.

Alternate routes

In our basic model, we assumed one preselected best route for every container move. However, the best route selection is a huge problem[22] by itself, given that numerous alternatives may exist for routing containers in multimodal transport. These arise from the many alternate origin ports, truck services, ship services, transshipment ports, and so forth. The optimal trade-off between time and cost is not always clear. For instance, one route may be faster and costlier, whereas another may be slower and cheaper. Then, selecting the best is not easy. We can easily incorporate such multiple, pareto-optimal routes in our formulation.

First, we take a route to mean any combination of transport paths and modes. We assume that we know the time and cost for each possible route. We define RDS_{ds} as the set of alternate routes between depot d and site s, RSD_{sd} as that between site s and depot d, and RSS_{mn} as that between sites m and n. Now, to extend the basic formulation, we simply break up each flow variable into several subvariables, one for each possible route. We add an index r to each variable to get the subvariable for each route as follows:

- x^r_{dmjk} ($j \le k$, $r \in RDS_{dm}$): the number of empty containers leaving depot d by route r at time t_j to reach site m at time t_k
- y^r_{nejk} ($j \le k$, $r \in RSD_{ne}$): the number of empty containers leaving site n by route r at t_j to reach depot e at t_k
- u^r_{smjk} ($j \le k$, $r \in RSS_{sm}$): the number of empty containers leaving site s by route r at t_j to reach site m at t_k
- v^r_{mnjk} ($j \le k$, $r \in RSS_{mn}$): the number of loaded containers leaving site m by route r at t_j to reach site n at t_k
- w^r_{mnjk} ($j \le k$, $r \in RSS_{mn}$): the number of leased (from the spot market) containers in v^r_{mnjk}

Note that each route can have its own distinct time and cost requirements. Using the above variables, we rewrite Eqs. 1, 2, and 4 in our formulation as

$$DI_{dk} = DI_{d(k-1)} + \sum_{s} \sum_{r \in RSD_{sd}} \sum_{j \le k} y^r_{sdjk} - \sum_{s} \sum_{r \in RDS_{ds}} \sum_{j \ge k} x^r_{dskj} \tag{1a}$$

$$\begin{aligned} SI_{sk} = SI_{s(k-1)} &+ \sum_{d} \sum_{r \in RDS_{ds}} \sum_{j \le k} x^r_{dsjk} + \sum_{n \ne s} \sum_{r \in RSS_{ns}} \sum_{j \le k} u^r_{nsjk} \\ &+ \sum_{m \ne s} \sum_{r \in RSS_{ms}} \sum_{j \le k} (v^r_{msjk} - w^r_{msjk}) - \sum_{d} \sum_{r \in RSD_{sd}} \sum_{j \ge k} y^r_{sdkj} \\ &- \sum_{m \ne s} \sum_{r \in RSS_{sm}} \sum_{j \ge k} u^r_{smkj} - \sum_{n \ne s} \sum_{r \in RSS_{sn}} \sum_{j \ge k} (v^r_{snkj} - w^r_{snkj}) \end{aligned} \tag{2a}$$

$$\begin{aligned} C = &\sum_{d} \sum_{s} \sum_{k} \sum_{r \in RDS_{ds}} \sum_{j \ge k} x^r_{dskj} XC^r_{dskj} \\ &+ \sum_{s} \sum_{d} \sum_{k} \sum_{r \in RSD_{sd}} \sum_{j \ge k} y^r_{sdkj} YC^r_{sdkj} \\ &+ \sum_{s} \sum_{n \ne s} \sum_{k} \sum_{r \in RSS_{sn}} \sum_{j \ge k} u^r_{snkj} UC^r_{snkj} \\ &+ \sum_{m} \sum_{n \ne m} \sum_{k} \sum_{r \in RSS_{mn}} \sum_{j \ge k} w^r_{mnkj} WC^r_{mnkj} \\ &+ \sum_{m} \sum_{n \ne m} \sum_{k} \sum_{r \in RSS_{mn}} \sum_{j \ge k} v^r_{mnkj} VC^r_{mnkj} \\ &+ \sum_{s} \sum_{k < K} h_s SI_{sk}(t_{k+1} - t_k) \end{aligned} \tag{4a}$$

Furthermore, the loaded container flows from the alternate routes must collectively meet each order's requirement:

$$\sum_{r \in RSS_{mn}} \sum_{j \le k} v^r_{mnjk} = NC_o \qquad t_k = DD_o \tag{5}$$

Equation 4a allows the loaded containers moving by alternate routes to have different costs and Eq. 5 allows them to start at different times from site m.

Finally, the number of leased containers cannot exceed the total number of loaded containers moving in each route r, so

$$w^r_{mnjk} \le v^r_{mnjk} \qquad j \le k,\ r \in RSS_{mn} \tag{6}$$

Time windows

Let EDD_o denote the earliest and LDD_o denote the latest due date for order o. As long as the loaded containers arrive within [EDD_o, LDD_o], the delivery for order o is acceptable to the destination site. We can address this in the same way as we addressed the alternate routes. Instead of one due date, we now postulate several possible due dates within [EDD_o, LDD_o]. The schedules for the possible transportation options would dictate these due dates. For instance, possible ship arrival times would dictate the times within the time window at which the containers may reach the destination site. If this is not possible, then one can arbitrarily pick some discrete arrival times within the range [EDD_o, LDD_o]. With this, the optimizer gets the freedom to choose one or more times to send the containers, and some containers may arrive earlier than LDD_o. This approach breaks up each order into multiple suborders, which increases the number of effective orders in our model. These suborders individually do not have fixed-order quantities, but they collectively satisfy the container requirement for an order. So if v_{mnjk} is the flow for a suborder with different t_k for an order o, then we have

$$\sum_{k \in TW_o} \sum_{j \le k} v_{mnjk} = NC_o \tag{5a}$$

where TW_o is the set of possible due dates in [EDD_o, LDD_o]. Furthermore, we have a constraint similar to Eq. 6:

$$w_{mnjk} \le v_{mnjk} \qquad j \le k \tag{6a}$$

Observe that the delivery of an order to the destination site may not be in one lot any more. The containers may arrive in separate lots at the different due dates during [EDD_o, LDD_o]. Requiring that the delivery be in a single lot within the time window will destroy the LP structure of the model because we would need to select a specific due date. However, we can achieve single-lot deliveries by defining a binary variable as follows:

$$\alpha_{ok} = \begin{cases} 1 & \text{if order } o \text{ is delivered in one lot at time } t_k \\ 0 & \text{otherwise} \end{cases}$$

Because the order can arrive at a single due date only, we have

$$\sum_{k \in TW_o} \alpha_{ok} = 1 \tag{7}$$

Now, to make the values of v-variables consistent with α_{ok}, we use the following:

$$v_{mnjk} = \alpha_{ok} NC_o \qquad k \in TW_o \tag{8}$$

Equation 6a still applies, but now the LP model becomes a MILP.

Container cleaning

For most tank containers, cleaning is imperative, unless they are dedicated to a specific product grade. If the sites do not have the cleaning facilities, then the sites must return emptied containers to depots for cleaning and cannot supply empty containers directly to other sites. Therefore, we delete all such events from the superlist. This eliminates all the u-variables from our formulation. Because all emptied containers must go to the depots for cleaning, there is no incentive for holding them temporarily at sites. Similarly, there is no incentive for getting clean containers at sites earlier than needed. This means that we can safely set $SI_{sk} = 0$. Then, Eq. 2 breaks up into two equations:

$$\sum_{d} \sum_{j \le k} x_{dsjk} = \sum_{n \ne s} \sum_{j \ge k} (v_{snkj} - w_{snkj}) \tag{2b}$$

$$\sum_{d} \sum_{j \ge k} y_{sdkj} = \sum_{m \ne s} \sum_{j \le k} (v_{msjk} - w_{msjk}) \tag{2c}$$

For the depots, we assume that the cleaning times are known and deterministic, and we know how many containers a depot can clean in a given time. In contrast to our basic formulation where the containers were always clean, we now need to keep stocks of both clean and unclean (or dirty) containers at the depots. To this end, we define DID_{dk} and DIC_{dk} to denote, respectively, the inventories of dirty and clean containers in depot d at time t_k. Let NC_{dk} be the actual number of containers that depot d cleans during [t_{k-1}, t_k], and NC_{dk}^U be the maximum number that it can clean during [t_{k-1}, t_k]. Knowing t_{k-1} and t_k, we can compute NC_{dk}^U a priori. With this, the balances for the dirty and clean containers are

$$DID_{dk} = DID_{d(k-1)} - NC_{dk} + \sum_{s} \sum_{j \le k} y_{sdjk} \tag{9a}$$

$$DIC_{dk} = DIC_{d(k-1)} + NC_{dk} - \sum_{s} \sum_{j \ge k} x_{dskj} \tag{9b}$$

Because a depot d cannot clean more containers than what it has and what is beyond its capacity, we have

$$NC_{dk} \le NC_{dk}^U \tag{10a}$$

$$NC_{dk} \le DID_{d(k-1)} \tag{10b}$$

Note that Eqs. 9a, 9b, and 10 will not force the dirty containers to be cleaned immediately on arrival, but only when clean containers are needed. This is not a problem because we can easily adjust the container inventories later to effect immediate cleaning. However, if that is not acceptable, then one can simply impose a small holding cost in the objective function for the dirty inventories at the depots to effect immediate cleaning. Furthermore, when the cleaning times are larger than the time intervals between successive events, the above equations do not exactly capture the system dynamics. We can develop alternate but somewhat more complex equations to handle cleaning in such a situation. However, for most practical purposes, we can cleverly adjust the numbers NC_{dk}^U to remedy the situation. Therefore, we have avoided presenting the more complex constraints to handle container cleaning.

Finally, we must add the cost of cleaning to the objective, as the cost can vary from depot to depot. Therefore,

$$\begin{aligned} C = &\sum_{d} \sum_{s} \sum_{k} \sum_{j \ge k} x_{dskj} XC_{dskj} + \sum_{s} \sum_{d} \sum_{k} \sum_{j \ge k} y_{sdkj} YC_{sdkj} \\ &+ \sum_{s} \sum_{n \ne s} \sum_{k} \sum_{j \ge k} u_{snkj} + \sum_{m} \sum_{n \ne m} \sum_{k} \sum_{j \ge k} v_{mnkj} VC_{mnkj} \\ &+ \sum_{m} \sum_{n \ne m} \sum_{k} \sum_{j \ge k} w_{mnkj} WC_{mnkj} + \sum_{d} \sum_{k} NC_{dk} CC_d \end{aligned} \tag{4b}$$

Multitype substitutable containers

In our discussion so far, we dealt with identical containers of a single type. In reality, many types of tank containers and chemicals exist. It may be possible to use several container types for some orders, whereas other orders may require specific container types. To extend our model to include multiple container types, let TC ($c = 1, 2, \ldots, TC$) denote the number of container types, V_c denote the volume capacity of type c, and SC_o denote the set of container types that are suitable for shipping an order o. We further assume that all containers, irrespective of their types, are of the same size, so the number of containers required for an order remains the same. Clearly, as we did earlier for alternate routes, we add an index c for the container type to each variable in our formulation. Thus, we now have inventory balance equations for each container type.

$$DI^c_{dk} = DI^c_{d(k-1)} + \sum_s \sum_{j \le k} y^c_{sdjk} - \sum_s \sum_{j \ge k} x^c_{dskj} \tag{1b}$$

$$SI^c_{sk} = SI^c_{s(k-1)} + \sum_d \sum_{j \le k} x^c_{dsjk} + \sum_{n \ne s} \sum_{j \le k} u^c_{nsjk} + \sum_{m \ne s} \sum_{j \le k} (v^c_{msjk} - w^c_{msjk}) - \sum_d \sum_{j \ge k} y^c_{sdkj} - \sum_{m \ne s} \sum_{j \ge k} u^c_{smkj} - \sum_{n \ne s} \sum_{j \ge k} (v^c_{snkj} - w^c_{snkj}) \tag{2b}$$

To ensure that each order's requirements are met, we use the following constraints analogous to Eqs. 5a and 6a:

$$\sum_{c \in SC_o} \sum_{j \le k} v^c_{mnjk} V_c \ge VC_o \tag{5b}$$

$$w^c_{mnjk} \le v^c_{mnjk} \qquad j \le k \tag{6b}$$

where VC_o is the total volume for order o. Note that $w^c_{mnjk} = v^c_{mnjk} = 0$ for $c \notin SC_o$. Finally, we rewrite the total cost as

$$C = \sum_c \Bigg[\sum_d \sum_s \sum_k \sum_{j \ge k} x^c_{dskj} XC^c_{dskj} + \sum_s \sum_d \sum_k \sum_{j \ge k} y^c_{sdkj} YC^c_{sdkj} + \sum_s \sum_{k<K} h^c_s SI^c_{sk}(t_{k+1} - t_k) + \sum_s \sum_{n \ne s} \sum_k \sum_{j \ge k} u^c_{snkj} UC^c_{snkj} + \sum_m \sum_{n \ne m} \sum_k \sum_{j \ge k} w^c_{mnkj} WC^c_{mnkj} + \sum_m \sum_{n \ne m} \sum_k \sum_{j \ge k} v^c_{mnkj} VC^c_{mnkj} \Bigg] \tag{4c}$$

Note that when we consider containers of different sizes, the above problem becomes a discrete MILP because we can no longer treat the v-variables as continuous. The model will retain its LP structure only if all the containers have the same size. In this case, Eq. 5b will become

$$\sum_{c \in SC_o} \sum_{j \le k} v^c_{mnjk} = NC_o \tag{5c}$$

Ship schedules and capacities

Ship schedules are implicit in our procedure for computing the occurrence times of various events. For instance, recall that we revised the due dates of orders (event time $t1$) to conform to the available ship schedules. Similarly, we computed $t2$ from $t1$ by using the estimated schedule of the ship that will carry the containers. Even the possible arrival times within the delivery time window are also dictated by the ship schedules. However, we assumed that ships had infinite capacities. In reality, operators call shippers in advance to check whether a target ship has the space to carry their containers. In many instances, changes take place even at the last minute, and the shipper may defer even some scheduled, but less important, containers to accommodate some rush and/or more important containers. Thus, predicting the space availability on a ship is not easy. In addition, a ship may visit several ports during its journey and pick up other containers or unload some containers. All these factors make the scheduling, a few months in advance, quite difficult. However, it is clear that limited space on ships and unforeseen ship delays can affect the container management decisions. Thus, what we present here is necessarily a simplified approach for handling ship schedules and finite ship capacities.

As stated earlier, our basic model assumes a specific ship for carrying each order. Therefore, during our event generation procedure, we can keep a record of the ships that will carry various orders. Because we know the number of containers for each order, we also know what each ship would carry and between which ports. Knowing the available space on each ship at various ports in its journey, we can identify all the overloaded ships and the orders that may not be able to travel on those ships. Clearly some of these orders must go either on the next available ship or on an earlier ship. This essentially means that we need to designate one or more alternate ships to carry these orders, which is nothing but allowing time windows for these orders. Thus, for all such orders, v_{mnjk} become variables, and Eqs. 5a and 6a apply.

Now, to ensure that we do not overload a ship at any point during its journey, we simply gather the v-variables for all the orders that the ship may carry during a port-to-port voyage, and force their sum to be less than the available space on the ship. Thus, in principle, it is easy to handle ship capacities in our formulation.

Nonuniform holding costs at depots

In practice, the holding costs at the depots may vary considerably with location. For example, if the operator rents space for storing the containers, then the rental charges may vary considerably with the depot location. Other operating costs, such as labor, energy, and insurance, may also vary. Furthermore, the cost of capital may also vary across regions. Therefore, the holding cost can be nonuniform across the depots. We can handle the holding costs at the depots by simply adding a cost term similar to the site holding cost to Eq. 4.

Two trade-offs can arise because of the nonzero holding cost at the depots and its nonuniformity. One is the trade-off between holding at sites vs. depots, and the other is the trade-off between holding at one depot vs. another in periods of lean demands. If the sites cannot hold or their holding costs are prohibitive, then the emptied containers would always immediately return to the depots. For the second trade-off, the optimal decision is not clear, and we can leave it to the optimizer.

Multisourcing

Multisourcing may be important, when the operator is the manufacturing company itself or its 4PL. The customer who requires certain product just places an order with a specific delivery due date, but does not specify the origin site. It is the responsibility of the operator to find the best source to supply this product and effect its delivery. The operator can now take a holistic view and find the best sources that minimize its total cost.

This is nothing but a special order with no prespecified origin site m. In the basic model, we know m a priori, and the v-variables are constants. In the present case, we do not know m, and must decide the best. Thus, we break up the entire order into various suborders, as we did earlier for the case of alternate routes and time windows. Now, we have a suborder for each possible origin site, and the corresponding order quantities become variables, and must sum to the total order quantity as follows:

$$\sum_{m} \sum_{j<=k} v_{mnjk} = NC_o \qquad t_k = DD_o$$

In-transit containers

Whenever the operator runs this scheduling model, some containers will always be in transit. These containers would reach their destination sites or depots at some known times during the scheduling period. Because the start times for these containers are negative, we can simply model them as if they started at time zero and will reach their destinations at the known times. We generate the appropriate arrival events for them in Phase 1 and generate all the events that may follow them in future. Then, depending on their origin sites or depots and their states (loaded, empty, clean, dirty, etc.), we can set the flow variables related to their arrival events at the known numbers of in-transit containers.

For instance, consider that NC empty containers are in transit at time zero from a depot d to reach site m at time t_k. We treat this simply as an order in which the supply depot is d. We generate all the future events (at t_k and beyond) for this order as usual, and generate one event at time zero corresponding to the departure of empty containers from depot d to site m. In other words, we set $x_{dm0k} = NC$, and eliminate (or fix as zero) all other x-variables and u-variables related to this order. At the other extreme, consider that NC containers are in transit at time zero from a site n to reach a depot e at t_k. In this case, we generate only one event at time zero for the departure of emptied containers from site n to reach depot e at t_k. In other words, we set $y_{ne0k} = NC$, and eliminate all other y-variables related to this order.

Nonzero loading/unloading times

Normally, the loading and unloading times are negligible compared to the site-to-site and depot-to-site transport times. In our model, we have subsumed (Figure 2) these times at various places. Figure 2 shows the delay components between various events for an order. As such, our model can easily handle the unloading and loading times, if we know a priori the numbers of containers involved. If this is not so, then we must incorporate some maximum loading/unloading times between appropriate time events. For instance, consider the case of NC loaded containers reaching a destination site. The site will unload them, and send them to some nearby depots. However, we do not know a priori how many containers will go to each depot. Therefore, we cannot know exactly the unloading time required for the containers going to a specific depot. Because this ignorance would prevent us from estimating the arrival time of the containers (Figure 2) at that depot, we assume that the site unloads all NC containers first, and then sends them to various depots. We make a similar assumption for loading at the origin sites. Although these assumptions do involve certain approximations in the cases of time windows and alternate routes, they are reasonable. In fact, the simple LP structure of the model would be in jeopardy, if we refuse to accept these as reasonable approximations.

Containers for storage

Chemical traders and even some chemical companies often use tank containers for mere short-term or long-term storage. For instance, an existing company dealing with phosgene-related products uses tank containers regularly for storage because the local regulations do not permit the company to use regular storage tanks for these products. Thus, a container company often serves orders that require no movement of loaded containers. Clearly, the origin and destination sites are the same ($m = n$) for such orders. In our formulation, we simply set $t1 = t2 + TSS(o, m, m)$ for such orders, where $TSS(o, m, m)$ is the duration for which the containers are used at the origin site m for storage. For such orders, we also set the transport cost of the loaded containers as zero. Also, note that the condition $m \neq n$ has to be removed appropriately from some summation terms in the objective function and constraints.

Revenue management

If the operator is a chemical company managing its own logistics, then it is obligated to serve all orders. It cannot refuse to serve an order regardless of its cost. Therefore, the objective of minimizing the total operation cost makes sense. However, for a 3PL/4PL company or a container company that serves spot orders along with contractual orders, the objective of maximizing profit makes more sense than that of minimizing cost. This is because serving a given order may not be attractive, if the cost for repositioning the empty containers to serve that order exceeds the revenue from the order. In fact, the operator would prefer to identify and refuse orders that reduce its profit. Thus, the preferred objective for such operators is to select and schedule orders that maximize gross profit. The business terminology for this is revenue or yield management.

To enable our model to perform yield management, let R_o denote the revenue per container from order o. Now we have two types of orders, mandatory and optional. The operator must serve the former, but would prefer to serve only the profitable ones from the latter. There is no change in the constraints for the former. However, v_{mnjk} becomes a variable with an upper

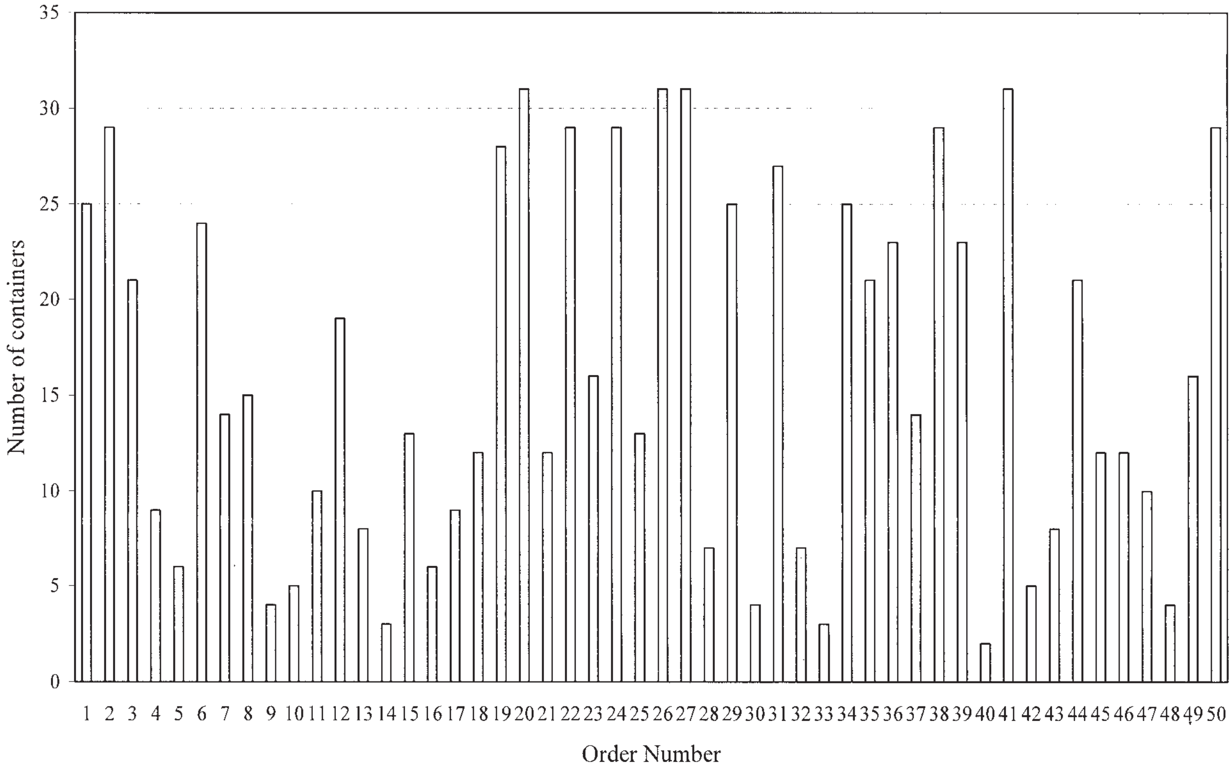

Figure 5. Empty container demands for 50 orders in Example 2.

bound of NC_o for the latter and Eq. 6a applies. Then, we can write the gross profit as

$$P = -\sum_d \sum_s \sum_k \sum_{j \geq k} x_{dskj} XC_{dskj} - \sum_s \sum_d \sum_k \sum_{j \geq k} y_{sdkj} YC_{sdkj} - \sum_s \sum_{n \neq s} \sum_k \sum_{j \geq k} u_{snkj} UC_{snkj} + \sum_m \sum_{n \neq m} \sum_k \sum_{j \geq k} v_{mnkj}(R_o - VC_{mnkj}) - \sum_m \sum_{n \neq m} \sum_k \sum_{j \geq k} w_{mnkj} WC_{mnkj} - \sum_s \sum_{k<K} h_s SI_{sk}(t_{k+1} - t_k) \quad (4d)$$

Maximizing P allows the optimizer to select only the profitable orders, but the selection could include even partial orders. However, the resulting formulation is still an LP. If partial orders cannot be accepted, then we must use a binary variable to force the selection of full orders only. For this, we define

$$\beta_o = \begin{cases} 1 & \text{if the operator serves an optional order } o \text{ fully} \\ 0 & \text{otherwise} \end{cases}$$

Now, $v_{mnjk} = \beta_o NC_o$ for each optional order o, and the formulation becomes a MILP.

Having discussed all the practically useful extensions of the model, we now consider three examples. Example 2 uses the basic simple model with no extensions, but illustrates a large industrial-scale scenario with many depots, sites, and orders. Example 3 illustrates the most important practical extensions on a smaller problem. Finally, Example 4 shows the application of our model to revenue management. However, note that our model, being an LP, can successfully solve large, industrial problems with all the extensions discussed above. It is only the lack of real problem data and the difficulty of exposition that prevented us from presenting a large example with all the extensions.

Example 2

To demonstrate the use of the basic model on a large problem, we selected arbitrary locations in China, and generated data using their latitude and longitude information. We randomly created 50 depots, 65 sites, and 500 orders to model a real scenario. For the sake of simplicity, we assumed truck as the sole mode of transport, even though the model can handle multimodal transport as discussed earlier. We used the standard "Great Circle" formula[23] and the relevant data from Choong et al.[10] to compute the distances between location pairs. Then, we used a basic transport cost (US$/km) and speed (km/h) to compute the transport times (h) and costs (US$) for all location pairs. However, we assumed a minimum transport time of 6 h, zero initial inventories at the sites, and a markup for the transportation cost of loaded containers. In this example, the transport costs for loaded containers varied from $100 to around $2000. Finally, we randomly generated the orders, their four attributes (origin site, destination site, number of containers, and due date), and initial inventories at the depots. Figure 5 shows the demands for 50 orders.

We used a FORTRAN program to generate the various events and times, the associated cost information, and the input files for GAMS 20.7. This example has a scheduling period of 824 h (34 days). Its LP involved 176,359 variables, 93,267 constraints, and 487,230 nonzeros. Its solution took 8538 iterations and 40.11 s of CPU time using CPLEX 7.5 on a Dell workstation with dual Intel Xeon 2.8-GHz processors and 2 GB

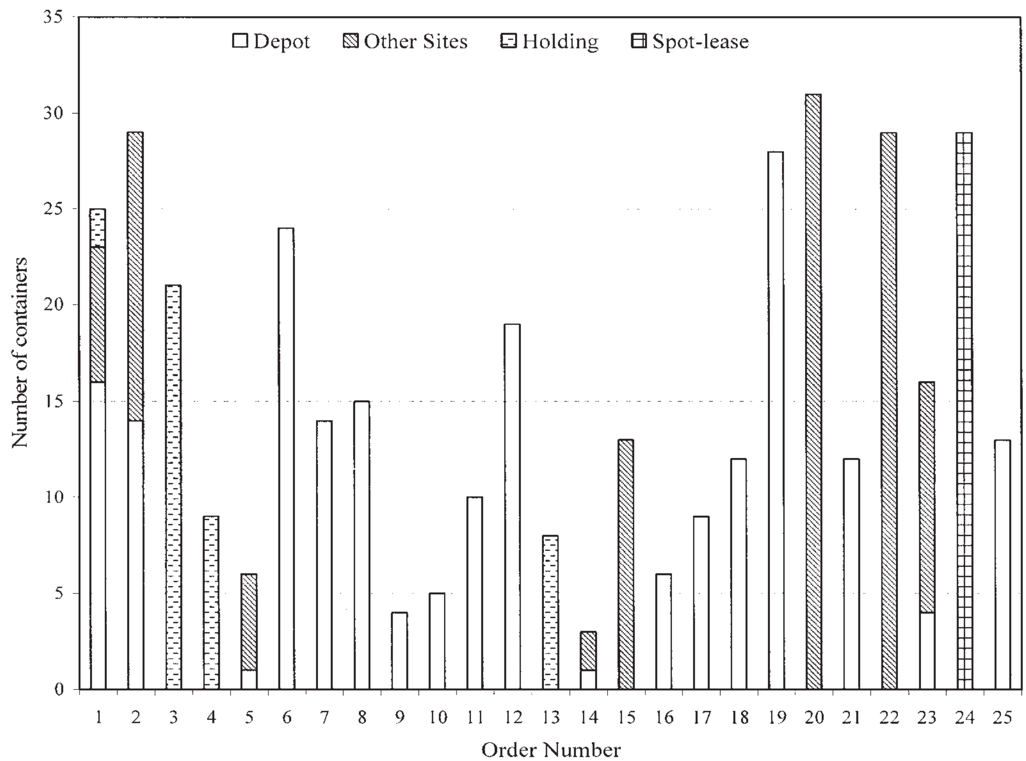

Figure 6. Fulfillment pattern for 25 orders in Example 2.

RAM. The presolve step of CPLEX 7.5 reduced the original LP drastically by eliminating 11,397 constraints and 15,677 variables, and making 60,262 substitutions. The reduced LP had 21,608 constraints, 100,420 variables, and 200,774 nonzeros, which represents reductions of roughly 77, 43, and 59% in constraints, variables, and nonzeros, respectively. The optimal cost for all container movements was \$7,440,912 (~ \$7.5 million).

Note that the 8245 containers moving over a month in this example are representative of the scale of operation of a typical big container company that moves about 100,000 containers per year or 8300 per month. Figure 6 shows the optimal order fulfillment pattern for 25 orders. We observe that the depots alone satisfy 260 orders (~ 50%), whereas the sites satisfy 126 orders (~ 25%). The operator needs to lease only a few (202 or ~ 2.5%) containers from the spot market. The depots and sites combine to satisfy most of the remaining orders.

Container use is a key concern in container management. One metric for this is the average time utilization of a container, which is the percentage time that a container spends on an average not idling in the inventories of depots or sites. Mathematically, we can write this as

$$\rho = \lim_{T\to\infty} \frac{1}{NT} \sum_{i=1}^{N} \int_0^T Z_i(u)du$$

where N is the number of containers in the system, T is the total operation time, and $Z_i(t) = 1$, if a container i is in use at time t, and zero otherwise. For this example with a finite time horizon, the above expression reduces to

$$\rho = 1 - \frac{\sum_{k=0}^{K-1} \left[(t_{k+1} - t_k)\left(\sum_d DI_{dk} + \sum_s SI_{sk} \right) \right]}{t_K \sum_d DI_{d0}}$$

The above gives us $\rho = 0.25$ (or 25%) for this example. The actual utilization will be slightly higher, because the containers do not have orders in this example toward the end of the scheduling horizon and they are simply waiting at the depots. In a dynamic situation, new orders will arrive to use them continuously during subsequent scheduling runs.

Because the orders demand 8245 containers and the operator leases only 202 containers from the spot market, the optimizer fulfills most orders by using and reusing the operator-owned 1260 containers. In other words, as we would expect, the optimizer uses a container more than once during the scheduling horizon. We define container turn as the number of times a container is used during the horizon. We can compute this metric of container use as

$$\tau = \frac{\text{Containers ordered} - \text{containers spot-leased}}{\text{Containers in the system}}$$

Note that we subtract the spot-leased containers in the above, given that the system uses a spot-lease container only once. For this example, $\tau \approx 6$. In other words, each container owned by the operator turns (up for work!) roughly (8245 − 202)/1260 or six times during the scheduling horizon. The higher the container turn, the more effective the use of the containers, although it is clear that this metric increases with the scheduling

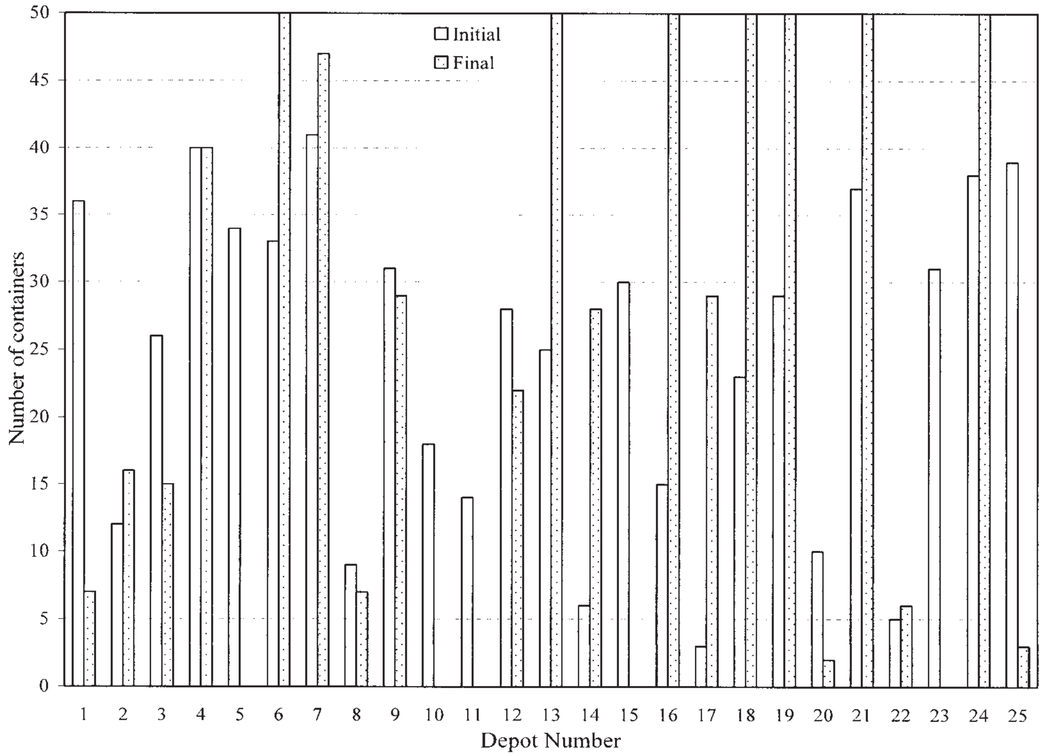

Figure 7. Initial and final inventories of containers at 25 depots in Example 2.

horizon. Therefore, a better metric would be the one that is normalized with respect to time (that is, container turn per unit time). We can compute this as τ/t_K. For this example, this is approximately 0.2/day or one turn every 5 days. One could also define other metrics such as cost per container or profit per container.

Figure 7 shows the initial and final inventories at the depots. As in Example 1, emptied containers prefer some depots more than others, and these depots easily reach their maximum capacity of 50 containers.

Example 3

In this example, we illustrate the use of our extended model with the features of time windows, ship schedules, tank container cleaning, and alternate routes. We consider 10 sites (S = 10), three depots (D = 3; D01, D02, and D03), three ports (P01, P02, and P03), 15 orders (O = 15), and a single type of tank containers that require cleaning after every use. We assume zonal coverage by ports. Thus, sites 1–4 use P01 as the origin port, sites 5–7 use P02, and sites 8–10 use P03. Normally, D01 serves sites 1–4, D02 serves sites 5–7, and D03 serves sites 8–10. However, each depot is free to serve any of the 10 sites. In this example, the granularity of time is 1 day.

Table 5 shows the order details. The early and late departure times in Table 5 are the earliest and latest times, respectively, at which the loaded containers can leave the origin site. Note that these time windows are different from those used in the model. The time for loading or unloading the containers is 1 day, so the empty containers must arrive for loading at the origin site 1 day before their scheduled departure time, and must stay 1 day for unloading at the destination site after their arrival. Table 6 lists the available ship services and their frequencies between various port-to-port pairs. For each pair, two alternate services are available. One is fast service (call it route r = 1) and the other is slow service (call it route r = 2). The cost per day for the fast route is three times the cost per day for the slow route, and the total cost is proportional to the transport time between the ports. The time required for export or import clearances at the ports is 1 day, which has been included in the total transport time between the origin site and destination site.

Although our model considered time windows at the destination site, it can easily handle those at the origin site as well.

Table 5. Order Data for Example 3

Order o	Origin Site m	Destination Site n	Departures from Origin Site: Early	Late	Containers NC_o
1	8	2	21	26	2
2	1	9	5	8	1
3	6	5	83	90	4
4	7	1	40	44	2
5	10	4	15	43	3
6	5	1	11	19	1
7	2	5	25	30	1
8	2	3	62	68	2
9	9	6	74	86	5
10	3	8	32	35	1
11	1	2	55	75	2
12	9	4	82	89	4
13	4	7	8	12	1
14	6	5	46	53	2
15	3	5	87	99	3

Table 6. Ship Services and Schedules for Example 3

Voyage	Ship Service	Departure Schedule	Transport Time (days)
P01 to P02	Fast	Every 4 days starting on day 3	4
	Slow	Every 6 days starting on day 2	6
P02 to P01	Fast	Every 7 days starting on day 5	4
	Slow	Every 5 days starting on day 1	6
P02 to P03	Fast	Every 3 days starting on day 6	4
	Slow	Every 6 days starting on day 4	6
P03 to P02	Fast	Every 5 days starting on day 11	4
	Slow	Every 10 days starting on day 5	6
P01 to P03	Fast	Every 8 days starting on day 10	5
	Slow	Every 10 days starting on day 7	11
P03 to P01	Fast	Every 12 days starting on day 4	5
	Slow	Every 8 days starting on day 10	11

To this end, we identify the possible times within these time windows at which the containers can depart from the origin sites for various orders. Because ships depart from the origin ports at specific times only, the empty container arrivals at the origin sites must match with the ship departure times. For instance, consider order 15. This order has a time window of [87, 99], and involves transport from P01 to P02. The possible ship departures around this time window for route 1 are 87, 91, 95, 99, and 103. Now, for a container to board a scheduled ship, it must reach the origin site sufficiently in advance so that (1) the site can load it, (2) the operator can move it from the origin site to the origin port, and (3) the port can perform export clearance. Because it takes 2 days to move the containers from site 3 to P01, and one day for export clearance at P01, it is clear that order 15 cannot depart on the ship at 87. Furthermore, to depart on the ships at 91, 95, 99, and 103, the containers must depart site 3 at 88, 92, 96, and 100. Only three of these—88, 92, and 96—are inside the time window. Therefore, only three of the 13 possible points in the time window match the ship schedule for route 1. In this way, we can identify all the possible times in the time window for each order o and route r (TW_{or}). Note that when an order may not involve transport by sea, then we must consider all points in its time window.

Now, as explained in the previous two examples, we construct the various input files for the GAMS program. In this example, we use the equations described under the extensions to alternate routes, time windows, and container cleaning. All depots can clean at most two containers per day ($NC^U_{dk} = 2$), but their cleaning costs are \$125 for D01, \$100 for D02, and \$200 for D03. The sites cannot clean containers, cannot hold containers except for loading or unloading, and cannot supply empty containers to each other. At time zero, there are 22 clean containers at the depots. The scheduling horizon is 130 days in this example. The LP model for this example had 3435 variables, 2216 constraints, and 11,372 nonzeros. GAMS/CPLEX 7.5 took 0.078 s and 320 LP iterations to solve the model, and the minimum cost was \$3921.

Table 7 summarizes the optimal departure times, supply depots, return depots, ship services and their departure times, and time windows for orders. Interestingly, the optimizer selects the fast routes for orders 1, 2, and 6 in spite of their higher costs. Note that the optimal departure times are not just the end times of the time windows, but somewhere in between for most orders. The flexibility of time windows enabled the optimizer to obtain a schedule that leases zero containers from the spot market. As illustrated earlier, this flexibility represents a key opportunity for reducing costs. Also, note that the zonal coverage by supply depots is mostly adhered to except for orders 5, 6, 9, and 11. However, most containers go to D02 for cleaning because D02 has the lowest cleaning cost, which seems to outweigh the transport cost. In fact, all of the 22 containers eventually end up at D02. Thus, returning to the designated supply depots is not optimal in this case. Orders 9 and 11 receive empty containers from two depots. Order 9 receives one container from D02 and four from D03, whereas order 11 receives one each from D01 and D02. The optimizer ships order 11 in two separate lots at two different times during the allowable departure time window. Table 8 recounts the events in a trip of a typical tank container. Note that, although D02 can clean up to two containers a day, it does not clean the container for two days. This is because there is no demand at that point in time, and thus it has no pressure to clean the container. The "pull" nature of our model forces container cleaning only when required. However, we can easily correct this behavior by shifting the cleaning activities forward post-solution.

Table 7. Optimal Supply/Return Depots, Departure Times from the Origin Sites, and Ship Services for Example 3

Order o	Supply Depots	Return Depots	Ship Service	Departure Times	Time Window
1	D03	D01	Fast	22	[21, 26]
2	D01	D03	Fast	7	[5, 8]
3	D02	D02	Slow	84	[83, 90]
4	D02	D02	Slow	40	[40, 44]
5	D01	D02	Slow	20	[15, 43]
6	D01	D02	Fast	14	[11, 19]
7	D01	D02	Slow	27	[25, 30]
8	D01	D02	Slow	64	[62, 68]
9	D02, D03	D02	Slow	82	[74, 86]
10	D01	D03	Slow	34	[32, 35]
11	D01, D02	D02	Slow	57, 59	[55, 75]
12	D03	D02	Slow	87	[82, 89]
13	D01	D02	Slow	8	[8, 12]
14	D02	D02	Slow	46	[46, 53]
15	D01	D02	Slow	89	[87, 99]

Example 3 has aptly illustrated all the key considerations of a real container management problem, and has yielded some results that may run contrary to practice.

Example 4

In this example, we use our model for revenue management instead of minimizing cost. We consider a problem with 10 mandatory orders ($o = 1, \ldots, 10$) and five optional orders

Table 8. Events in One Trip of a Typical Container in Example 3

Day	Event Description
5	Empty container starts from D01 toward S01
6	Arrives at S01 and starts loading
7	Loaded container departs S01 for S09 by the fast service from P01 to P03
18	Reaches S09 from P03 and begins unloading
19	Empty, dirty container departs S09 for D02
22	Reaches D02 from S09, ready for cleaning
24	Clean container is available at D02

Table 9. Order Information for Example 4

Order o	Order Type	Origin Site m	Destination Site n	Departure Time from m	Containers NC_o
1	Mandatory	8	2	21	2
2	Mandatory	1	9	5	1
3	Mandatory	6	5	83	4
4	Mandatory	7	1	40	2
5	Mandatory	10	4	15	3
6	Mandatory	5	1	11	1
7	Mandatory	2	5	25	1
8	Mandatory	2	3	62	2
9	Mandatory	9	6	74	5
10	Mandatory	3	8	32	1
11	Optional	1	2	55	2
12	Optional	9	4	82	2
13	Optional	4	9	104	3
14	Optional	6	10	46	2
15	Optional	3	7	87	3

(o = 11, . . . , 15). Table 9 shows the order details. As in example 3, we take 10 sites and three depots. For the sake of simplicity, we do not allow time windows and alternate routes in this example. The cleaning costs are \$40 for D01, \$50 for D02, and \$60 for D03. At time zero, D01 has 10 clean containers, D02 has four, and D03 has eight.

The LP for this example had 1198 variables, 1121 constraints, and 3603 nonzeros. GAMS/CPLEX 7.5 took 0.031 s and 78 iterations to solve this LP. The maximum profit is \$836. Table 10 shows the actual costs, revenue, and profit for each order. Note that the optimizer had no choice but to serve some nonprofitable mandatory orders. However, it did exercise its freedom to refuse the nonprofitable optional order 14. We listed order 14 in Table 10 merely to show its nonprofitability, although it is not in the optimal solution.

Now suppose that D01 has 10 clean containers at time zero, D02 has eight, and D03 has four. For this case, the optimizer gives a maximum profit of \$716, and serves order 14 in spite of its negative margin. This is because D03, the lowest cost depot for the origin site 9 for order 9, has insufficient containers at the time of departure of order 9. By serving order 14, the optimizer moves containers to D03 to fulfill order 9. Apparently, this is a cheaper option than sourcing the containers from other depots or the open market. In the absence of an advanced methodology, such as the one presented in this paper, an operator would normally decline to take a seemingly nonprofitable order 14. However, the correct decision is contrary, given that order 14 in fact increases the overall profit, when considered together with all the orders. Clearly, human decisions based on simple considerations, as routinely taken by the container operators, can mislead, and this example aptly illustrates the utility of our model in this respect.

Conclusions

We have presented an important problem of tank container management in global chemical logistics, which has received little attention in the literature. The two-step, event-based, demand-driven, deterministic methodology presented in this paper is novel and simple, and accommodates most of the relevant real-life considerations. The ability to handle order-specific and time-dependent information is a major advantage of our approach. Because it involves an LP whose size substantially reduces during the preprocessing step of a commercial LP algorithm such as CPLEX, it helps solve swiftly the large, industrial-scale problems. This is extremely important because, in most cases, it is impossible to predict precisely all future shipment orders. Changes, additions, cancellations, and the like in orders and data always occur, and the user must repeatedly generate schedules using the updated information in an extremely dynamic business environment. For example, it is common for ship schedules to change at short notice. A simple way to address the change in ship schedules is to resolve the entire problem by incorporating the new ship schedule information in the model and taking into account the present status of the empty/loaded container positions. In this case, the formulation will not change, even if some customers cancel their earlier orders. However, the best way to address uncertainties is through stochastic formulations. One approach is stochastic programming that would still keep the formulation an LP, but would require solving a series of LPs. The other popular approach is robust optimization, where the goal is to find a "good" solution that would be robust enough or a solution that would be insensitive to uncertainties in the input parameters. We hope to address this need in a future communication.

A tacit assumption in our work is that the demands are for integral numbers of containers. This is invariably true in the case of tank containers, and is unlike the case of dry containers,

Table 10. Costs, Revenues, and Profits for Orders in the Optimal Solution to Example 4

Order o	Revenue (\$)	Transport and Cleaning Costs (\$ per order)				Profit (\$)
		Depot-to-Site	Site-to-Site	Site-to-Depot	Cleaning	
1	1440	150	860	200	80	150
2	495	50	330	75	60	−20
3	1200	200	850	200	200	−250
4	1170	120	780	100	80	90
5	2295	225	1530	300	130	110
6	540	165	360	50	40	−75
7	630	100	420	50	50	10
8	750	200	500	100	80	−130
9	2250	525	1500	250	250	−275
10	630	50	420	75	60	25
11	900	100	400	200	80	120
12	1512	150	960	120	80	202
13	2268	180	1260	225	180	423
14	1060	100	820	150	120	−130
15	2106	150	1170	180	150	456

where a demand of less-than-container-load may occur. Even in this case, the operators would consolidate the demands in such a way that the final requirements would still be integral numbers of containers. Because our problem is strategic in nature, it is not important to consider how the contents of such a consolidated container are delivered to its many customers.

Finally, it is noteworthy that our methodology also applies directly to the management of the widely used dry containers with minor modifications. We believe that the proposed methodology is more compact and uses fewer variables than the existing network modeling approach for managing dry containers. We expect this to result in a computational advantage for our approach. However, we hope to report more on this comparison in a future communication.

Acknowledgments

The authors acknowledge the financial support for this work under a Strategic Research Program Grant 00221050030 from the Agency for Science, Technology, and Research (A*STAR), Singapore. This is a joint project between The Logistics Institute—Asia Pacific, National University of Singapore, and The Logistics Institute, Georgia Institute of Technology, Atlanta, GA. The authors are grateful to Juan Carlos Morales, Prof. Martin Savelsbergh, and Prof. Alan Erera of Georgia Institute of Technology for some useful discussions, comments, analyses, and data exchanges during this work.

Notation

Indices

c = type of container
d, e = depot (supply, return)
j, k = position in the superlist of event times
m, n = site (origin, destination)
o = order
r = alternate transport route
s = site

Sets

RDS_{ds} = alternate routes between depot d and site s
RSD_{sd} = alternate routes between site s and depot d
RSS_{mn} = alternate routes between site m and site n
SC_o = container types that are suitable for shipping order o
TW_o = possible delivery times in the time window for order o

Parameters

CC_d = cost of cleaning one container in depot d
D = number of depots
DD_o = time at which loaded containers are due at the destination site for order o
DI_d^U = maximum number of containers that depot d can hold at any time
EDD_o = earliest due time at the destination in the time window for order o
LDD_o = latest due time at the destination in time window for order o
h_s = cost of holding one empty container at site s ($/h)
K = number of event times in the superlist
NC_o = number of loaded containers required by order o
O = number of orders
R_o = revenue per container from order o
S = number of sites or customer locations
SI_s^U = maximum number of containers that site s can hold at any time
$t1$, $t2$, $t3$, $t4$, $t5$ = times at which various events happen for an order (Figures 2 and 3)
t_k = kth time in the superlist of event times
TC = number of container types available
TDS = time between the departure of containers from a depot and the arrival at a site
TSD = time between the arrivals of containers at a site and then at a depot
TSS = time between the arrivals of containers at an origin site and then at a destination site
XC_{dskj} = cost of moving one empty container from depot d at time t_k to reach site s at time t_j
YC_{sdkj} = cost of moving one empty container from site s at time t_k to reach depot d at time t_j
UC_{snkj} = cost of moving one empty container from site s at time t_k to reach site n at time t_j
V_c = volume of container type c
VC_o = total volume of cargo for order o
WC_{mnkj} = total cost of leasing one container from the spot market at site m and time t_k, and returning it after reaching site n at time t_j

Superscript

U = upper limit

Variables

C = total cost of moving all containers
DI_{dk} = number of containers in depot d at time t_k
DIC_{dk} = number of clean containers in depot d at time t_k
DID_{dk} = number of dirty containers in depot d at time t_k
NC_{dk} = number of containers that a depot d cleans during $[t_{k-1}, t_k]$
SI_{sk} = number of empty containers at site s at time t_k
x_{dmjk} = number of empty containers moving from depot d at t_j to reach site m at t_k
y_{nejk} = number of empty containers moving from site n at t_j to reach depot e at t_k
u_{smjk} = number of empty containers moving from site s at t_j to reach site m at t_k
v_{mnjk} = number of loaded containers moving from site m at t_j to reach site n at t_k
w_{mnjk} = number of empty containers spot-leased from the open market in v_{mnjk}

Greek letters

α_{ok} = 1, if the loaded containers for order o arrive in one lot at t_k and 0 otherwise
β_o = 1, if the operator fills an optional order o fully and 0 otherwise
ρ = average time utilization of containers
τ = container turn

Literature Cited

1. Karimi IA, Srinivasan R, Han PL. Unlock supply chain improvements through effective logistics. *Chemical Engineering Progress*. 2002;5: 32-38.
2. Reddy PCP, Karimi IA, Srinivasan R. A novel solution approach for scheduling crude oil operations. *AIChE Journal*. 2004;50:1177-1197.
3. Jetlund AS, Karimi IA. Improving the logistics of a fleet of multi-parcel chemical tankers. *Computers & Chemical Engineering*. 2004; 28:1267-1283.
4. White WW. Dynamic transshipment networks: An algorithm and its application to the distribution of empty containers. *Networks*. 1972;2: 211-236.
5. Florez H. *Empty Container Repositioning and Leasing: An Optimization Model*. PhD Thesis. New York, NY: Polytechnic Institute of New York; 1986.
6. Crainic TG, Gendreau M, Dejax P. Dynamic and stochastic models for the allocation of empty containers. *Operations Research*. 1993;41: 102-126.
7. Shen WS, Khoong CM. A DSS for empty container distribution planning. *Decision Support Systems*. 1995;15:75-82.
8. Abrache J, Crainic TG, Gendreau M. A new decomposition algorithm

for the deterministic dynamic allocation of empty containers. Publication CRT-99-49. Montreal, Quebec, Canada: Centre de Recherche sur les Transports, Université de Montréal; 1999:1-20.
9. Cheung RK, Chen CY. A two-stage stochastic network model and solution methods for the dynamic empty container allocation problem. *Transportation Science*. 1998;32:142-162.
10. Choong ST, Cole MH, Kutanoglu E. Empty container management for intermodal transportation networks. *Transportation Research Part E—Logistics and Transportation Review*. 2002;38:423-438.
11. Chen C, Chen S. A dynamic container allocation model. Working paper. Taipei, Taiwan: Department of Transportation Management, National Cheng Kung University; 1993.
12. Chen C, Ma K. A stochastic dynamic multicommodity container allocation model. Working paper. Taipei, Taiwan: Department of Transportation Management, National Cheng Kung University; 1995.
13. Lai KK, Lam K, Chan WK. Shipping container logistics and allocation. *The Journal of the Operational Research Society*. 1995;46:687-697.
14. Del Castillo E, Cochran JK. Optimal short horizon distribution operations in reusable container systems. *Journal of the Operational Research Society*. 1996;47:48-60.
15. Lutzebauer M. *Systems with Return Logistics*. Frankfurt am Main, Germany: Deutscher Fachverlag GmbH; 1993.
16. Kroon L, Vrijens G. Returnable containers: An example of reverse logistics. *International Journal of Physical Distribution & Logistics Management*. 1995;25:56-68.
17. Lim MF, Karimi IA. A slot-based formulation for single-stage, multiproduct batch plants with multiple orders per product. *Industrial and Engineering Chemistry Research*. 2003;42:1914-1924.
18. Ku HM, Karimi IA. Scheduling in serial multiproduct batch processes with finite interstage storage: A mixed integer linear program formulation. *Industrial and Engineering Chemistry Research*. 1988;27:1840-1848.
19. Gupta S, Karimi IA. Scheduling a two-stage multiproduct process with limited product shelf life in intermediate storage. *Industrial and Engineering Chemistry Research*. 2003;42:2365-2380.
20. Bazaara MS, Jarvis J. *Linear Programming and Network Flows*. New York, NY: Wiley; 1977.
21. Brooke A, Kendrick D, Meeraus A, Raman R. GAMS: A user's guide. *GAMS Development Corporation*, Washington, DC; 1998.
22. Ng YJ. *Stolt Tank Containers—Best Route Solution*. Dual-degree MS Thesis. Singapore: The Logistics Institute—Asia Pacific, National University of Singapore; 2003.
23. Chen FY, Goh M, Lee J, Ou J, Sharafali M, Teo CP, Goh PG, Sivanathan P. ST logistics: Distributing consumer goods in China. *International Transactions in Operational Research*. 2001;8:203-219.

Manuscript received Dec. 5, 2003, and revision received May 18, 2004.