Journal of the Operational Research Society (2002) 53, 415–426 0160-5682/02 $15.00

www.palgrave-journals.com/jors

Stowage planning in maritime container transportation

J-G Kang[1] and Y-D Kim[2]*

[1]*Swiss Federal Institute of Technology-Lausanne (EPFL), Lausanne, Switzerland; and* [2]*Korea Advanced Institute of Science and Technology, Daejon, Korea*

We consider a stowage-planning problem of arranging containers on a container ship in the maritime transportation system. Since containers are accessible only from the top of the stack, temporary unloading and reloading of containers, called shifting, is unavoidable if a container required to be unloaded at the current port is stacked under containers to be unloaded at later ports on the route of the ship. The objective of the stowage planning problem is to minimize the time required for shifting and crane movements on a tour of a container ship while maintaining the stability of the ship. For the problem, we develop a heuristic solution method in which the problem is divided into two subproblems, one for assigning container groups into the holds and one for determining a loading pattern of containers assigned to each hold. The former subproblem is solved by a greedy heuristic based on the transportation simplex method, while the latter is solved by a tree search method. These two subproblems are solved iteratively using information obtained from solutions of each other. To see the performance of the suggested algorithm, computational tests are performed on problem instances generated based on information obtained from an ocean container liner. Results show that the suggested algorithm works better than existing algorithms.

Journal of the Operational Research Society (2002) **53**, 415–426. DOI: 10.1057/palgrave/jors/2601322

Keywords: stowage planning; heuristics; sea transport

Introduction

In the modern maritime transportation system, minimizing transportation time and costs while maximizing container ship utilization is considered as a main objective. Transportation time and costs in the maritime container transportation are affected largely by the loading and unloading operations at ports. On average, a container ship spends approximately 60% of her time at the ports.[1] Therefore, a significant amount of cost and time savings can be obtained through an effective and efficient planning of loading/unloading operations or cargo handling at the ports.

As can be seen in Figure 1, the cargo space of a container ship is made up of a number of *bays*, collections of stacks of containers along the length of a ship. Generally, each bay in the cargo space is divided into *above-deck* and *below-deck* by *hatch covers*, and sub-areas of a bay divided by hatch covers are called *holds*. Each hold is composed of a group of *stacks*, and each stack is composed of vertically arranged groups of *cells*. Each cell is a physical location or a slot where a container is to be loaded. Containers loaded below-deck can be unloaded only after all containers loaded above-deck on the hatch cover above are removed as well as the hatch cover.

In the ocean cargo industry, container ships make repeated tours of a series of ports according to their planned routes. At each port on a tour of a container ship, containers are unloaded and additional containers destined for subsequent ports are loaded. Time duration required for loading and unloading depends on the arrangement of the cargo on board the ship, ie the *stowage plan*, which specifies where each container is loaded on the ship. Stowage plans, if not prepared well enough, may cause unnecessary handling time, time required for temporary unloading and re-loading of containers from/onto the ship, called *shifting*, and for movements of gantry cranes at the ports. Consequently, port efficiency and ship utilization are largely affected by stowage plans.

In general, shifting is caused by *overstowage*, which denotes the situation when containers that should be unloaded at the current port are placed under other containers that should go further in the ship's route. In this case, the latter containers should be temporarily unloaded in order for the former containers to be unloaded at the current port. These temporarily unloaded containers, commonly referred to as *overstows*, must then be re-loaded before the ship leaves.

On the other hand, movements of gantry cranes are affected by the distribution of containers on the container

Correspondence: Y-D Kim, Department of Industrial Engineering, Korea Advanced Institute of Science and Technology, Yusong-gu, Daejon 305-701, Korea.
E-mail: ydkim@mail.kaist.ac.kr

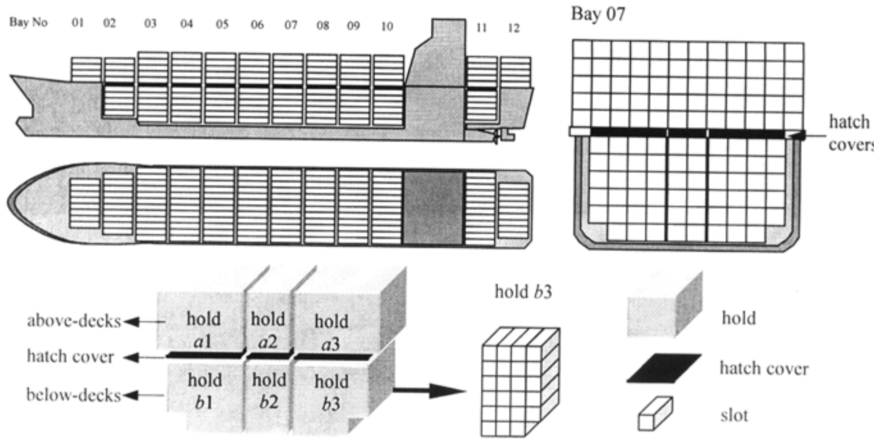

Figure 1 Cargo spaces of a typical modern container ship.

ship, since the number of cranes that can be assigned to a ship at each port is limited. If containers with the same destination are spread over the container ship in different bays, it takes a longer time to unload the containers than in the case in which those containers are placed near each other, since more crane movements are necessary. Delays caused by such crane movements can be avoided if containers with the same destination are stowed in the same bay.

There have been a few studies on the *stowage planning problem*, the problem of finding a stowage plan with a certain objective. Botter and Brinati[2] give another integer programming model considering various characteristics related to stowage operations, such as constraints related to the ship's stability, crane movements and loading sequence of the containers, and use an implicit enumeration method to solve it. Since it is not easy to solve a stowage planning problem optimally in a reasonable amount of time, various heuristic algorithms have been developed for the objective of minimizing the number of required shifting operations or maximizing the number of containers that can be loaded on a ship. For example, Shields[3] suggests a somewhat simple solution algorithm, in which a number of different possible loading plans are randomly generated and the best is selected. Avriel and Penn[4] develop a heuristic procedure, called the whole columns heuristic, in which integer programs are solved after preprocessing of data. Later, Avriel *et al*[5] treat the stowage planning problems as a two-dimensional stacking problem, and give a heuristic procedure, called the suspensory heuristic procedure, for the objective of minimizing the number of shifting operations. However, they assume that there is only one large bay in a ship without considering constraints related to hatch covers and the stability of the ship. Recently, Wilson and Roach[6,7] present a tabu search method combined with a branch and bound search and packing heuristic based on the conceptual processes employed by human planners. On the other hand, Saginaw and Perakis[8] and Shin and Nam[9] develop rule-based decision support systems or expert systems that are based on the knowledge of a manager or an operator in charge of loading and unloading operations.

Most existing heuristic methods have drawbacks in that they do not consider the stability of the ship, crane movements or stowage plans at subsequent ports. Although various practical constraints in the stowage planning problem are considered in Shields' algorithm, it does not guarantee the optimality of the solutions, and it may take a long computation time to find a reasonably good solution. On the other hand, the integer programming approach guarantees the optimality only if the integer programs can be solved, but they are too large to be solved in a reasonable time even for small-sized problems.

This study focuses on the stowage planning problem with the objective of minimizing the time that container ships spent in port terminals, or equivalently, a weighted sum of the number of shifting operations and the frequency of required crane movements, which are major factors that influence the time. For the problem, we develop a heuristic solution method in which the problem is divided into two subproblems, one for assigning container groups to the holds and one for determining a loading pattern of containers assigned to each hold. The former subproblem is solved by a greedy heuristic based on the transportation simplex method, while the latter is solved by a tree search method. These two subproblems are solved iteratively using information obtained from solutions of each other.

Problem statement

The stowage planning problem considered in this paper is that of finding an arrangement plan of the cargo on the ship that minimizes time required for loading and unloading while maintaining ship's stability. We consider the loading/unloading time at all ports included in a tour of the ship. Loading/unloading time increases if there are more shifting operations caused by overstows or if there are more longitudinal movements of cranes. Moving distance or time required for the movements is proportional to the number of bays where containers are to be loaded on or unloaded from. Thus, in order to minimize loading/unloading time, we need to minimize overstows as well as the number of bays occupied by the containers with the same destination port.

In general, exact loading or unloading time varies depending on the locations of the containers being handled and the performance of the cranes, and so does the time required for shifting and time required to move a gantry crane. However, such time duration, especially the time duration required at later ports, cannot be estimated exactly when a stowage plan is made. In this study, therefore, each of those time durations is assumed to be constant and given, and the average time durations for those operations (loading, unloading, shifting and crane movement) are used for the performance measure to evaluate stowage plans. Here, we do not consider time required for loading or unloading operations that are not needed for shifting, since they are required anyway, regardless of the stowage plan. Also, we normalize those times for simplicity of the model by letting the average time for shifting a container be a unit time as follows. Let T_S and T_C be the average time required for shifting a container and the average time required for moving a gantry crane between two bays. Then, in the objective function of the stowage planning problem, the time required for shifting a container is set to 1 and the time for moving a crane from one bay to another is set to $c \equiv T_C/T_S$.

To assure the stability of a container ship, a stowage plan should satisfy several constraints. A ship becomes unstable if the vertical, transverse or longitudinal distribution of the ship's weight is excessively unbalanced. Some stowage plans may result in the instability of the ship. In these cases, changes of the stowage plans, ie rearrangements of containers, are necessary to regain the ship's stability. While the ship's stability is affected by various factors, we consider the following three most influencing factors in this study: metacentric height (GM), heel and trim.

The *metacentric height* (*GM*) of a ship is defined as the distance between the centre of gravity (G) and the metacentre (M) of the ship as illustrated in Figure 2(a). For a ship to be stable, GM must be greater than the minimum allowable metacentric height of the ship. Otherwise, the ship will capsize. To make GM greater, heavier containers should be placed at lower positions. Making GM greater may conflict with the objective of minimizing the number of shifting operations, if the heavier containers are to be transported to nearer destinations. The *heel* is the inclination of a ship resulting from turning the ship in the direction of starboard or port (Figure 2(a)). The heel of a ship with respect to the centreline must be zero, or at least within a very narrow range around zero. The *trim* of a ship is the

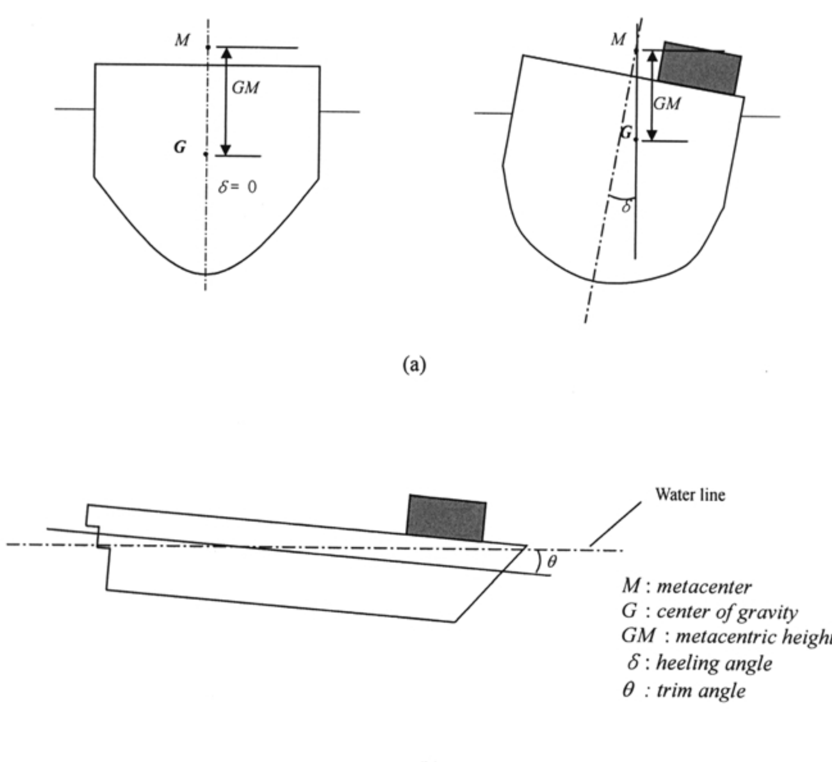

Figure 2 Stability of a ship. (a) GM and heel, (b) trim.

difference in draft forward and aft of the ship (Figure 2(b)). It should also be close to zero or at least within certain pre-specified limits for good performance of the ship. The stability constraints stated above (*GM*, *heel* and *trim*) can be linearized by representing them as the moments resulting from the composition of all weights of the cargo with respect to the vertical, transverse and longitudinal coordinates.

In this study, the following assumptions are made.

1. Containers to be delivered on the container ship are of the same size (40 ft standard containers).
2. The route of the ship's tour is given.
3. The number of containers to be delivered from one port to another is known (at the starting port, ie the port at which the stowage plan is to be made) for all pairs of ports included in the tour.
4. The initial stowage pattern of the ship at the starting port is given.
5. The number of containers to be loaded does not exceed the capacity of the ship at any port.
6. The unit cost (time) related to crane movements is the same at all ports, and the time required to handle an overstow is the same at all ports.

Solution approach

In the solution procedure suggested in this paper, the stowage-planning problem is partitioned into two subproblems, the problem of assigning container groups to holds and that of determining specific positions or slots for the containers assigned to each hold. Here, a container group denotes a set of containers with the same source (port of loading: POL), the same destination (port of destination: POD), and the same weight. Throughout this paper, the former subproblem will be denoted as the *G-to-H (groups to holds) problem*, while the latter will be called the *C-to-S (containers to slots) problem*.

These two subproblems are solved one after the other iteratively using the information obtained from solutions of each other. With this iterative procedure, we may take account of the interdependency of the two subproblems. For example, if a solution of the C-to-S problem results in overstowage at an iteration, the number of containers that can be stowed in the same hold is limited to a certain number for container groups that cause the overstowage to avoid such overstowage at the next iteration. As a result, the procedure will generate a different stowage plan, possibly a feasible plan.

Now, we give a detailed description of the solution methods for the subproblems. For notational simplicity, ports are indexed according to the order they are visited on a given route of a ship. For example, port 3 denotes the third port visited on the route of the ship.

Solution method for the G-to-H problem

Since a stowage plan at a port is affected by those made at previously visited ports, one has to make stowage plans at all ports included in a tour of container ships simultaneously. However, it may not be possible to make the plans simultaneously because the problem becomes too large (there are too many variables and constraints) if they are considered together. In the suggested method, the G-to-H problem is defined at each port separately and solved separately. The G-to-H problem at a port is defined and solved based on solutions of the G-to-H problems at the ports to be visited earlier. To cope with the interdependency of the stowage plans made at different ports, we consider not only the time required to handle containers at the current port but also time required to handle the containers (to be loaded at the current port) at later ports to unload them in the G-to-H problem. In this study, the G-to-H problem at a port (say, port k) is formulated as a mathematical programme, which is similar to that of the fixed charge transportation problem. First, we give the notation used in the formulation.

Indices

- i index for container groups, $i = 1, \ldots, I$
- j index for holds, $j = 1, \ldots, J$
- k index for ports, $k = 1, \ldots, K$
- q index for bays, $q = 1, \ldots, Q$

Parameters (given)

- D_i number of group-i containers to be loaded at the current port
- H_q set of holds included in bay q
- B_j bay in which hold j is included
- e_j number of empty slots in hold j
- E_q number of empty slots in bay q, ie $E_q = \sum_{j \in H_q} e_j$
- W_i weight of a group-i container
- X_j longitudinal coordinate of the centre of hold j
- Y_j transverse coordinate of the centre of hold j
- Z_j vertical coordinate of the centre of hold j
- l_L lower limit of the longitudinal moment of the ship
- u_L upper limit of the longitudinal moment of the ship
- u_T upper limit of the transverse moment of the ship
- u_V upper limit of the vertical position of the centre of gravity of the ship
- c (normalized) time required to move a crane from one bay to another

Parameters (to be estimated)

- a_{ij} number of shifting operations required to be performed at the current port if a group-i container is assigned to hold j
- b_{ij} number of shifting operations to be incurred at later

ports if a group-i container is assigned to hold j at the current port

d_{ij} time required for crane movements at the POD of group-i containers if group-i containers are assigned to hold j at the current port

Decision variables

x_{ij} the number of group-i containers allocated to hold j

$$y_{ij} = \begin{cases} 1 & \text{if a group-}i\text{ container is allocated to hold } j \\ & (\text{if } x_{ij} > 0) \\ 0 & \text{otherwise} \end{cases}$$

$$z_q = \begin{cases} 1 & \text{if there is a container allocated to bay } q \\ & (\text{if } \sum_i \sum_{j \in H_q} y_{ij} > 0) \\ 0 & \text{otherwise} \end{cases}$$

In this study, values for a_{ij}, b_{ij} and d_{ij} are estimated as follows using the information on the current locations of the containers.

Estimation of a_{ij}

Case 1. If hold j is an above-deck hold, a_{ij} is set to 0.
Case 2. If hold j is a below-deck hold, a_{ij} is set to N_j, where N_j is the number of containers loaded in the above-deck hold over hold j.

Estimation of b_{ij}

Case 1. If hold j is an above-deck hold: then (let hold j' be the below-deck hold under hold j)
Case 1a. If there is no container in hold j' whose POD is nearer than the POD of group-i containers, b_{ij} is set to n_N/n_T, where n_N is the number of empty slots above the containers whose PODs are nearer than the POD of group-i containers and n_T is the number of empty slots in the hold;
Case 1b. If there are containers in hold j' whose PODs are nearer than the POD of group-i containers, b_{ij} is set to 1.
Case 2. If hold j is a below-deck hold, b_{ij} is set to n_N/n_T.

Estimation of d_{ij}

Case 1. If there is no container in bay B_j whose POD is the same as the POD of group-i containers, then d_{ij} is set to c.
Case 2. If there are containers in bay B_j whose PODs are the same as the POD of group-i containers, then d_{ij} is set to 0.

Note that a_{ij} denotes the number of shifting operations for other containers caused by container group i, while b_{ij} denotes the number of shifting operations for group-i containers caused by other containers. Figure 3 gives three examples to show how to estimate a_{ij} and b_{ij}. In all the examples, we have $a_{ij} = 0$ and $a_{ij'} = 8$ because $N_{j'} = 8$, ie eight containers are loaded in hold j, the above-deck hold over hold j'. In the example given in Figure 3(a), b_{ij} is set to 2/8 since $n_N = 2$ and $n_T = 8$ in hold j (Case 1a), while $b_{ij'}$ is set to 0 because $n_N = 0$ in hold j' (Case 2). In Figure 3(b), b_{ij} is set to 1 (Case 1b) and $b_{ij'}$ is set to 1 because $n_N = n_T$ in hold j' (Case 2). Also, in Figure 3(c), b_{ij} is set to 1 (Case 1b) and $b_{ij'}$ is set to 2/6 because $n_N = 2$ and $n_T = 6$ in hold j' (Case 2).

Now, we give an integer linear programme for the G-to-H problem at port k.

$$[\text{GH}_k] \quad \text{Minimize } \sum_i \sum_j a_{ij} y_{ij} + \sum_i \sum_j b_{ij} x_{ij} + c \sum_q z_q + \sum_i \sum_j d_{ij} y_{ij} \tag{1}$$

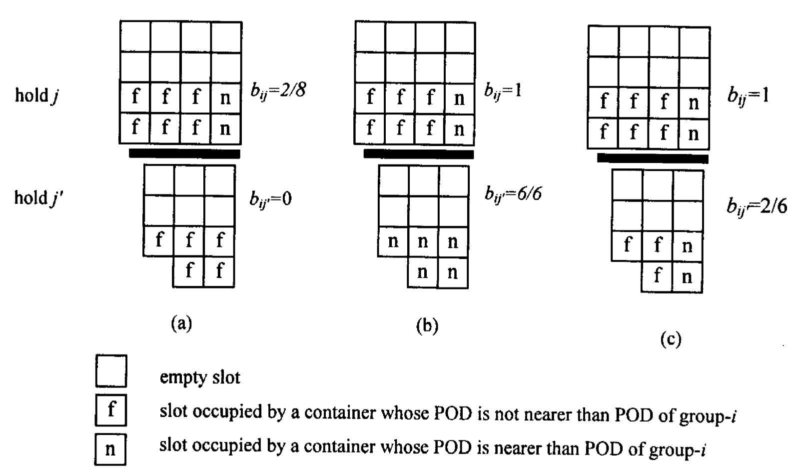

Figure 3 Examples for the estimation of a_{ij} and b_{ij}.

subject to

$$\sum_j x_{ij} = D_i \qquad \forall\, i \qquad (2)$$

$$\sum_i x_{ij} \leqslant e_j \qquad \forall\, j \qquad (3)$$

$$x_{ij} \leqslant e_j\, y_{ij} \qquad \forall\, i, j \qquad (4)$$

$$\sum_i \sum_{j \in H_q} y_{ij} \leqslant E_q\, z_q \qquad \forall q \qquad (5)$$

$$\sum_i \sum_j W_i\, Z_j\, x_{ij} \leqslant u_V \qquad (6)$$

$$-u_T \leqslant \sum_i \sum_j W_i Y_j x_{ij} \leqslant u_T \qquad (7)$$

$$l_L \leqslant \sum_i \sum_j W_i\, X_j\, x_{ij} \leqslant u_L \qquad (8)$$

$$x_{ij} \geqslant 0 \text{ and integer} \qquad \forall i, j \qquad (9)$$

$$y_{ij} \in \{0, 1\} \qquad \forall i, j \qquad (10)$$

$$z_q \in \{0, 1\} \qquad \forall q \qquad (11)$$

The objective function to be minimized denotes the (estimated) handling time required to handle overstows at the current port k and subsequent ports k' for $k' > k$ and to move crane(s) at port k and subsequent ports k'. Constraint (2) ensures that all containers are loaded and constraint (3) ensures that the number of containers loaded in hold j does not exceed the number of available slots in hold j. Constraints (4) and (5) are added to define relationships among decision variables, that is, $y_{ij} = 1$ if $x_{ij} > 0$, and $z_q = 1$ if $y_{ij} > 0$ for any hold $j \in H_q$. Constraints (6), (7) and (8) ensure the ship's stability in terms of the GM, heel and trim, respectively.

Problem [GH$_k$] is similar to the fixed charge transportation problem (FCTP), but it differs from a typical FCTP in that there are additional constraints related to the ship's stability and an additional fixed charge related to variable z_q. Although there are various methods to solve FCTPs as surveyed by Adlakha and Kowalski[10] and Palekar *et al*,[11] those methods cannot be directly applied to [GH$_k$] because of such differences. Since it is difficult to solve [GH$_k$] optimally in a reasonable time, we suggest a heuristic. This heuristic follows a general procedure of a heuristic developed by Gilbert and Madan[12] for production planning and scheduling problems.

In the heuristic suggested here for the G-to-H problem, we employ a general procedure of the transportation simplex method. In this heuristic, an initial solution is obtained in a form of the basic feasible solution of the transportation problem after the objective function of [GH$_k$] is approximated to a linear form. Then the solution is improved with a method similar to the pivot operation used in the transportation simplex method. A detailed description of the algorithm is given below.

Obtaining an initial solution. First, the objective function is approximated to a linear form using new cost coefficients $\hat{a}_{ij}$ defined as $\hat{a}_{ij} = a_{ij} + d_{ij} + ce_j/E_{q'}$ for all i and j, where $q' = B_j$. Note that time for crane movement, c, is allocated to each hold in proportion to the ratio of the number of empty slots in the hold to the total number of empty slots in the bay. With this approximation, integer variables, y_{ij} and z_q, related to the fixed charge portion of the objective function are eliminated. Also, constraints (4) and (5) can be eliminated with such an approximation. Then, [GH$_k$] is relaxed into a linear program, [LGH$_k$] as follows.

$$[\mathrm{LGH}_k] \qquad \text{Minimize} \sum_i \sum_j (\hat{a}_{ij}/e_j + b_{ij})x_{ij} \qquad (1')$$

Subject to (2), (3), (6), (7), (8) and

$$x_{ij} \geqslant 0 \qquad \forall i, j \qquad (9')$$

In this study, [LGH$_k$] is solved optimally with a commercial software package. The optimal solution of [LGH$_k$], whether it is feasible to [GH$_k$] or not (because the integrality constraint is not satisfied), is modified to a form of a basic feasible solution of the transportation problem with constraints (2) and (3), ie a solution in which at most $(I + J - 1)$ variables have positive values. The procedure of modifying solutions of [LGH$_k$] follows a general procedure for constructing an initial basic feasible solution of a transportation problem such as the northwest corner rule. In the procedure, we use a transportation tableau with cells, rows and columns corresponding to variable x_{ij}, constraint (2), and constraint (3), respectively. Note that, in the general procedure for the transportation problem, the value of a basic variable is set to be large enough to exactly use up the remaining supply in its row or the remaining demand in its column (whichever is smaller), and this row or column is eliminated from further consideration.

In the suggested procedure, among variables that do not lie in an eliminated row or column, we select a variable (x_{ij}) with a positive value (in the solution of [LGH$_k$]) that is equal to (or closest to) the remaining supply in its row or the remaining demand in its column. Then, either a row or column associated with such a selection is eliminated from further consideration. This procedure is continued until there is only one variable that can be chosen, when both the row and the column associated with the variable are eliminated.

If there still are variables with positive values in cells not selected yet after all rows and columns are eliminated, pivot operations are performed after identifying a chain reaction or a θ-loop in the transportation tableau.[13,14] In other words, values of the variables associated with these cells are reduced to 0 and values of the variables associated with the cells in the θ-loop are changed so that the resulting solution has a form of the basic feasible solution of the transportation problem. On the other hand, if there is a row or column not eliminated yet and none of the variables in that row or column can have positive values, one of the variables in that row or column is selected randomly and it is considered as a basic variable.

If constraints (6)–(8) are not satisfied by the solution resulting from the above procedure, the solution is modified with a method that is similar to the pivot operation of the transportation simplex method as follows. Using the transportation tableau, the basis is updated by replacing an entering variable for a leaving variable. In this method, we select as an entering variable a non-basic variable that will reduce the number of violated constraints with the minimum increase in the objective function value if entered into the basis with a pivot operation. Note that a leaving variable can be identified for each non-basic variable, which is a candidate entering variable, as in the transportation simplex method. The variable selected as the leaving variable is a basic variable with the smallest value among those corresponding to donor cells in the θ-loop that includes the entering variable in the transportation tableau. Such pivoting operations are repeated until the solution becomes feasible.

Improving the solution. Next, the initial solution is improved with a method that is similar to the pivot operation of the transportation simplex method as follows. Define the fixed charge portion of the objective function of $[\mathrm{GH}_k]$, $\sum_i \sum_j a_{ij}\, y_{ij} + c \sum_q z_q + \sum_i \sum_j d_{ij}\, y_{ij}$ as $f(x_{ij})$. Note that the value of $f(x_{ij})$ is easily determined by x_{ij} from the definition of y_{ij} and z_q. Using the transportation tableau as in the method for obtaining an initial feasible solution, the basis is updated by replacing an entering variable for a leaving variable. Although pivot operations to obtain new basic feasible solutions are the same as those of the above method and of the simplex method, the criterion used for selection of an entering variable is different. Here, we select as an entering variable a non-basic variable that will give the maximum decrease in the objective function (1′) and will violate none of constraints (6), (7) and (8) as follows.

For each non-basic variable, assuming it is the entering variable, one can identify a leaving variable as in the transportation simplex method. Then, the change in the objective function value of $[\mathrm{GH}_k]$ due to replacement of the leaving variable with the entering variable can be computed as $\Delta V_{ij} = \sum_i \sum_j a_{ij}\, \Delta x_{ij} + \Delta f(x_{ij})$, where Δx_{ij} is the change of the value of x_{ij} (or the value of the entering variable) after the replacement and $\Delta f(x_{ij})$ is the change in the value of $f(x_{ij})$ due to the replacement. Here, $\Delta f(x_{ij})$ can be computed as $\Delta f(x_{ij}) = \sum_i \sum_j a_{ij}\, \Delta y_{ij} + c \sum_{B_i} \Delta z_q + \sum_i \sum_j d_{ij}\, \Delta y_{ij}$, where Δy_{ij} and Δz_q are the changes of the values of y_{ij} and z_q due to the changes of values of x_{ij} in the θ-loop. Note that the values of y_{ij} and z_q can be easily computed from the definition of the variables.

Since a solution obtained with this pivoting operation should not violate constraints (6), (7) and (8), we have to check the feasibility of the solution resulting from the pivot for each candidate entering variable. In the suggested algorithm, a non-basic variable can be considered as a candidate entering variable if all of the following three conditions are satisfied:

$$\text{(C1)} \quad \sum_i \sum_j W_i\, Z_j\, (x_{ij} + \Delta x_{ij}) \leqslant u_V$$

$$\text{(C2)} \quad -u_T \leqslant \sum_i \sum_j W_i\, Y_j\, (x_{ij} + \Delta x_{ij}) \leqslant u_T$$

$$\text{(C3)} \quad l_L \leqslant \sum_i \sum_j W_i\, X_j\, (x_{ij} + \Delta x_{ij}) \leqslant u_L$$

where Δx_{ij} is the value of the changes in the basic and non-basic (entering) variables resulting from the pivot operation to make the variable currently being considered a new basic variable. Although one can compute the maximum change in the value of the candidate entering variable that will not violate the constraints and use that value for pivoting, such a method is not employed in the suggested algorithm. This is because by doing so the resulting solution would not have the property of a basic feasible solution of the transportation problem with constraints (2) and (3). Note that the general concept of the suggested algorithm is to improve the solution using pivoting operations of the transportation simplex method.

In the suggested algorithm, pivoting operations are repeated until the solution cannot be improved any more. The following summarizes the solution procedure for $[\mathrm{GH}_k]$ described above.

Procedure 1 (*Solution method for* $[\mathrm{GH}_k]$)

Step 1. Find an optimal solution of [LPR].

Step 2. Modify the solution in such a way that the solution has a form of the basic feasible solution of the transportation problem with constraints (2) and (3).

Step 3. For each non-basic variable, identify the θ-loop in the transportation tableau and compute Δx_{ij}. For each non-basic variable x_{ij} that satisfies conditions C1, C2 and C3, compute ΔV_{ij}.

Step 4. If $\Delta V_{ij} > 0$ for all non-basic variables, stop. Otherwise, obtain a new solution by performing a pivoting operation to bring into the basis a non-basic variable with the minimum value of ΔV_{ij} and go back to step 3.

Solution method for the C-to-S problem

After the G-to-H problems are solved for all k, containers allocated to each hold are assigned to available slots of the hold, that is, the C-to-S problem is solved, for the objective of minimizing overstows throughout a tour of the ship. In the C-to-S problem, loading plans (or stowage plans) can be determined for each hold independently of other holds, since which groups of containers should be loaded on and unloaded from each hold at each port were determined by the solution of the G-to-H problem. Therefore, it is sufficient to determine at which slots containers are loaded (in each hold at each port). However, those loading plans at a port should be made considering the loading/unloading plans of the other ports. Here, the stability of the ship is not considered, since it was considered in the G-to-H problem.

Generally, containers with the same port of loading (POL) and the same port of destination (POD) are stowed at adjacent slots in a hold. Therefore, if relative horizontal positions among container groups are determined, the stowage pattern of the containers can be easily determined because relative positions of containers in a column can be determined by the following simple rules. If different containers are to be loaded in the same column, a container of a further POD is loaded under the others. If two or more containers have the same destination, a heavier container is loaded under the others.

In the algorithm suggested in this paper, an implicit enumeration method is employed to find the best stowage pattern (in terms of overstows). Since stowage plans can be determined for each hold separately, this enumeration method is not expected to require excessively long computation time. A tree search method is used to enumerate all possible configurations of the containers, ie relative positions among the container groups, and to find the best one. In the search tree, nodes represent relative positions among the container groups in a hold and levels of the nodes represent POLs at which containers related to the nodes are loaded.

In the suggested algorithm, we add a dummy container group to fill all empty slots if the number of containers assigned to a hold is less than the total number of empty slots of the hold. Since there are various methods to allocate the containers to the slots satisfying the relative positions among them, exact positions of the containers cannot be specified by only determining the relative positions among the container groups without such dummy containers.

Figure 4 gives a simple example that shows the loading pattern representation scheme used in this study. In the figure, (a) shows a tree drawn with the representation scheme and (b) shows the physical loading pattern corresponding to the tree. The numbers in the parentheses in each node of the tree represent relative positions among the container groups and the level of a node represents the port where containers are loaded. Node (0,2,4) at level 1 of the tree (Figure 4(a)) represents a configuration in which at port 1, containers of POD 2 are located on the left of containers of POD 4, and there are empty slots to the left of the containers of POD 2 (Figure 4(b)). In addition, node (3,5) at level 2 represents that at port 2, containers of POD 3 and containers of POD 5 are loaded in slots where there were no containers and where there were containers that were unloaded at port 2.

Overall solution procedure

A solution obtained by solving the G-to-H problem and the C-to-S problem sequentially may not be very good or there may be many overstows in some cases. This is because the information on loading plans at subsequent ports is not considered when solving the G-to-H problem. Overstows can be reduced if such information is utilized when solving the G-to-H problem by not assigning too many containers to a hold at which overstowage will occur at subsequent ports. In the suggested algorithm, the two problems are solved iteratively so that such information can be taken into account.

Suppose that at an iteration of the iterative procedure overstowage occurs because container group i is loaded over another container group whose POD is further than that of group i. In this case, the coefficient (b_{ij}) in problem [GH$_k$] at the POL of group i is modified to avoid assigning group-i containers to the same hold at the next iteration. Then the procedure restarts solving the G-to-H problems from the G-to-H problem of the port at which group-i containers are loaded, to save computation time.

The iterative procedure is summarized below. In the procedure, k' denotes the port at which the occurrence of overstowage is being checked and k'' denotes the POL of containers that cause overstowage.

Procedure 2. *(Algorithm for the stowage planning problem)*

Step 0. Set $k'' = 1$ and $k' = 2$.

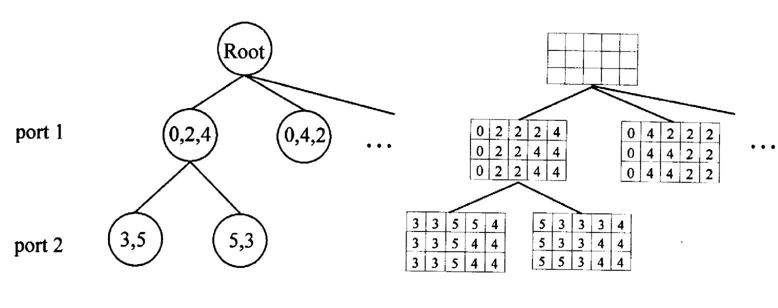

Figure 4 Tree search method. (a) A tree, (b) the loading pattern associated with the tree.

Step 1. Solve the G-to-H problems, $[GH_k]$ for $k = k'', k''+1, \ldots, K$.

Step 2. Solve the C-to-S problem for each hold at ports $k'', k''+1, \ldots, K$ using the solutions of the G-to-H problems.

Step 3. If overstowage occurs at port k', go to step 5. Otherwise, go to step 4.

Step 4. Let $k' = k'+1$. If $k' > K$, stop. Otherwise, go to step 3.

Step 5. If containers causing the overstowage were loaded at port k'', go to step 6. Otherwise, go to step 7.

Step 6. Re-define $[GH_{k'}]$, after letting $b_{ij} = b_{ij} + n_O/n_X$, where n_O is the number of shifting operations required by the overstowage and n_X is the number of group-i containers assigned to hold j at port k''. Solve $[GH_k]$ for $k = k'', k''+1, \ldots, K$, and solve the C-to-S problem for each hold at ports $k'', k''+1, \ldots, K$ using the solutions of the G-to-H problems. If the resulting solution is improved, let $k' = k''+1$, and go to step 1. Otherwise, go to step 7.

Step 7. Let $k'' = k''+1$. If $k'' = k'$, let $k' = k'+1$ and $k'' = 1$, and go to step 3. Otherwise, go to step 5.

Computational experiments

To investigate the performance of the iterative procedure suggested in this paper, computational tests were done on 180 randomly generated problems, 10 problems for each of all combinations of three levels for the number of ports (4, 6 and 8), three levels for the capacity of the ship (2500, 3000 and 4000 TEUs), and two levels for the number of container weight groups (2 and 3). These parameter values used for problem generation were selected so that resulting test problems reflect real stowage planning problems in a Korean maritime transportation company relatively well. It is assumed that ships with the capacities of 2500, 3000 and 4000 TEUs have 12, 15 and 20 bays, respectively. Each bay of the ships consists of three hatch covers and six holds (as illustrated in Figure 1). Also, in problems in which containers are of two different weights (heavy or light), a heavy container was assumed to be twice as heavy as a light one. In these problems, the number of heavy containers and that of light containers were set to be approximately equal. On the other hand, in problems with three container weight groups (heavy, medium and light), the ratio of the weights of the containers was assumed to be 5:3:1. Specific weights of containers are not considered here, since it is assumed that containers can be loaded on a ship up to the ship's capacity (in TEUs), regardless of the weights of the containers, once the stability condition is satisfied.

The number of containers that are to be transported from port i to port j was generated from $DU(0.8c, 1.2c)$, where $DU(a,b)$ denotes the discrete uniform distribution with range $[a, b]$ and c denotes the average number of containers that are shipped between two ports. Here, c was set to be equal to the average number of containers on board the ship divided by $n(n-1)/2$, where n is the number of ports to be visited in a tour of the ship. The average number is set to 95% of the ship's capacity in the test problems. Since it is assumed that the number of containers to be loaded does not exceed the ship's capacity as stated earlier, the value of c is modified (reduced) if the value obtained with the above method makes the ship's capacity constraint violated. In addition, it is assumed that certain containers are already loaded on the ship at the beginning of a tour. Destinations, weights and locations (in the ship) of those containers were generated randomly, but in such a way the resulting problem reflect real situations relatively well.

To evaluate the performance of suggested algorithm, we compared each of the two parts of the suggested algorithm with a method modified from the algorithm of Botter and Brinati[2] for solving the G-to-H problem and the suspensory heuristic method (SH) developed by Avriel *et al*[5] for solving the C-to-S problem, denoted hereafter by MBB and SH, respectively. In MBB, the G-to-H problem was solved by assigning container groups to holds using the model of Botter and Brinati, while containers are assigned to slots directly in the original model of Botter and Brinati. Note that the original model cannot be used for the stowage planning problem considered in this paper because it requires excessive computation time. As mentioned earlier, SH can deal with the stowage planning problem for a ship with one rectangular bay only, and thus the slot for each container is determined individually without taking account of hatch covers and the ship's stability.

As it is very difficult to obtain optimal solutions, performance of the algorithm was shown with a relative performance measure, the percentage reduction of the solution (the time required for loading and unloading containers throughout a given tour of a ship) of each algorithm from a benchmark solution. Here, the solution value obtained from the combination of MBB and SH was used as the benchmark solution. In cases where a solution of the MBB-SH combination could not be found after a sufficiently long time, say 24 hours, the solution of the combination of the suggested algorithm for the G-to-H problem and SH was used for the benchmark solution. The computational tests were done on a personal computer with a 500 MHz Pentium III processor, and CPLEX 6.5 was used to solve [LPR] and integer programs for MBB.

Results of the tests are given in Tables 1 and 2, which give average and standard deviation of the percentage reduction of the solutions from the benchmark solutions for each set of 10 problems defined by the parameter combinations. The tables also show the number of problems for which each algorithm found the best solutions. In the tables, the algorithms suggested in this research for the G-to-H problem and C-to-S problem are denoted by

Table 1(a) Performance of the algorithms in easier (smaller) problems. Percentage reduction from the solution of the (MBB, SH) combination

			MBB				*H-Simplex*			
			TS		*SH*		*TS*		*SH*	
Weight levels	*Ports*	*Bays*	*Avg.*	*St. Dev.*	*Avg.*	*St. Dev.*	*Avg.*	*St. Dev.*	*Avg.*	*St. Dev.*
2	4	12	1.8	10.2	0	—	17.8	31.6	0.6	23.4
		15	18.9	30.6	0	—	10.6	24.3	−2.3	33.7
		20	5.6	9.4	0	—	20.1	22.4	9.5	29.0
	6	12	3.4	16.2	0	—	47.8	22.0	40.1	29.4
		15	13.9	13.9	0	—	42.8	42.8	40.0	40.0
		20	9.8	21.4	0	—	43.3	34.7	43.2	33.7
3	4	12	9.1	10.0	0	—	22.1	33.2	12.7	38.1
		15	8.5	10.5	0	—	13.8	26.7	6.7	28.2
		20	12.1	10.0	0	—	22.8	18.9	12.9	18.7
Overall			9.2	14.7	0	—	26.8	28.5	18.2	30.5

Table 1(b) Performance of the algorithms in easier (smaller) problems. The frequency each algorithm found the best solution

			MBB		*H-Simplex*	
Weight levels	*Ports*	*Bays*	*TS*	*SH*	*TS*	*SH*
2	4	12	7	4	9	6
		15	8	4	6	5
		20	6	5	8	6
	6	12	0	0	8	2
		15	1	0	9	4
		20	0	1	10	8
3	4	12	5	0	4	2
		15	3	1	5	3
		20	2	0	6	2
Total			32	15	65	38

Table 2(b) Performance of the algorithms in harder (larger) problems. The frequency each algorithm found the best solution

			H-Simplex	
Weight levels	*Ports*	*Bays*	*TS*	*SH*
2	8	12	9	5
		15	8	6
		20	6	8
3	6	12	8	2
		15	7	3
		20	8	2
	8	12	7	3
		15	7	5
		20	6	6
Total			66	40

Table 2(a) Performance of the algorithms in harder (larger) problems. Percentage reduction from the solution of the (H-Simplex, SH) combination

			H-Simplex			
			TS		*SH*	
Weight levels	*Ports*	*Bays*	*Avg.*	*St. Dev*	*Avg.*	*St. Dev.*
2	8	12	5.2	7.6	0	—
		15	10.9	9.9	0	—
		20	3.3	9.4	0	—
3	6	12	6.5	9.9	0	—
		15	6.3	10.9	0	—
		20	4.5	7.6	0	—
	8	12	5.5	10.1	0	—
		15	7.5	9.6	0	—
		20	2.3	12.2	0	—
Overall			5.8	9.7	0	—

H-Simplex (heuristic based on the transportation simplex method) and TS (tree search method), respectively.

As can be seen in the tables, H-Simplex and TS work better than MBB and SH, respectively. TS gave solutions that are approximately 9% (6% in larger problems) better than those from SH on average, although SH gave better solutions than TS in some cases. A reason for such out-performance of TS over SH may be that TS finds the best relative horizontal positions by checking all possible alternatives with an implicit enumeration method, although simple (but good) rules are used for determining relative vertical positions of containers in a column. On the other hand, SH uses heuristic rules for determining a loading pattern, ie vertical as well as horizontal positions of the containers.

The difference in the performance of the algorithms for the G-to-H problem is more significant. Solutions from H-

Simplex were approximately 18% better than those from MBB. This may be because H-Simplex takes account of the possibility that a stowage plan at the current port may cause the overstowage in the subsequent ports to find out which holds result in the minimum overstows in all ports. Also, H-Simplex uses the information obtained from solutions of the C-to-S problems to improve its solution, while MBB solves the G-to-H problem independently without considering such information. Note that the overall iterative procedure was applied to H-Simplex only, since it could not be easily applied to MBB because of characteristics of the algorithm of Botter and Brinati.[2]

Table 3 shows the average computation time required for each problem to solve a problem. H-Simplex required much shorter computation time than MBB, especially in larger problems. MBB could not find a feasible solution for larger problems after several hours. Although it is not shown in the table, in some cases the MBB-SH combination could not solve a problem even after a few days. Since the stowage planning problems should be solved relatively frequently, it should not take such a long time to solve a problem. The computation time required for the suggested algorithm (H-Simplex with TS) does not seem to increase exponentially although it contains an implicit enumeration method, TS. Note that TS does not search all possible positions of individual containers but checks all possible *relative* positions among container groups that are assigned to a hold. In addition, the number of groups assigned to a hold at each port is not too large, and hence all possible relative positions among the groups can be examined in a reasonable amount of time.

Although it is not reported as a table here, the sign tests, non-parametric statistical tests for paired experiments, were done to see the difference in the performance of the four combinations of the algorithms. Results of the tests showed that there were statistically significant (in the significance level of 0.01) differences in the performance between all pairs of the four combinations. Considering the computation time required for the algorithms tested and qualities of the solutions obtained from those algorithms, it can be argued that the suggested heuristic solution method is significantly better than existing algorithms and it is a viable tool for stowage planning in the maritime container transportation.

Table 3 Average CPU seconds required for each problem

			MBB		*H-Simplex*	
Weight levels	*Ports*	*Bays*	*TS*	*SH*	*TS*	*SH*
2	4	12	57.5	51.9	32.0	35.7
		15	87.8	67.2	29.0	20.6
		20	138.0	124.5	36.2	31.3
	6	12	4247.8	4246.3	110.1	118.9
		15	5819.5	5819.2	122.8	120.3
		20	24 580.4	24 550.1	121.8	122.3
	8	12	*	*	365.5	366.5
		15	*	*	381.7	380.9
		20	*	*	401.8	398.1
3	4	12	78.5	69.5	42.8	43.8
		15	184.4	174.3	115.4	112.0
		20	283.5	264.2	228.5	222.3
	6	12	*	*	94.5	90.4
		15	*	*	188.3	140.3
		20	*	*	447.3	425.8
	8	12	*	*	525.5	495.3
		15	*	*	620.7	543.0
		20	*	*	640.5	592.0

*Solutions were not obtained within 24 h of CPU time.

Concluding remarks

In this study, we considered a stowage-planning problem for a container ship with the objective of minimizing cargo-handling time at ports, or equivalently, minimizing the number of shifting operations and crane movements at the ports on the route of the ship. Because of the computational complexity of the problem, the problem was decomposed into two subproblems, one for assigning groups of containers to holds of the ship and one for determining loading patterns of containers in each hold. The first subproblem was formulated as a set of integer programmes and solved by a heuristic algorithm that employed a general procedure of the transportation simplex method, while a tree search method was suggested for the second. These two subproblems were solved iteratively using information obtained from solutions of each other. Computational experiments showed that the suggested algorithm worked better than a commercial software package for integer programming using a model modified from that of Botter and Brinati[2] and the suspensory heuristic procedure of Avriel *et al*.[5]

Although we considered several aspects or characteristics of cargo handling operations at ports to make the research practical, there are still more to be done. In this study, we considered only one standard type of containers, but there may be containers of various sizes. In such cases, additional constraints should be taken into account. For example, if there are two types of containers (40 ft and 20 ft containers) are handled, possible positions of containers are limited by the sizes of the containers. Since a container is supported on its four corner points, it must be placed in positions where either the ship or other containers provide suitable supports. In general, a 40 ft container can be loaded on two 20 ft containers, but two 20 ft containers cannot be loaded on a 40 ft container. Because of this requirement, it is not desirable to have containers with different lengths be stowed in the same stack.

In addition, one has to consider a case in which positions for special cargos such as refrigerated containers and dangerous cargos are limited to a certain area, consider other types of constraints related to the stability of a ship, or consider other objectives such as balancing the workloads of the cranes. Also, one needs to determine sequences of unloading or loading containers from or onto a container ship that can minimize travel distances of cargo-handling equipment for an efficient operation of ports or ships. In fact, it is necessary to determine a stowage plan and loading sequences simultaneously for the maximum efficiency, because the operations of quay cranes and material handling equipment in a yard are closely related to the stowage plan. In some cases, intentional shifting may be needed to reduce shifting operations at subsequent ports. Therefore, we have to check whether the overall number of shifting operations can be reduced if currently loaded containers are temporarily unloaded, and re-loaded after containers to be loaded at the current port is loaded.

References

1 Peterkofsky RI and Daganzo CF (1990). A branch and bound solution method for the crane scheduling problem. *Transport Res* **24B**: 159–172.

2 Botter RC and Brinati MA (1992). Stowage container planning: a model getting an optimal solution. *IFIP Trans* **B-5**: 217–229.

3 Shields JJ (1984). Container ship stowage: a computer-aided preplanning system. *Marine Technol* **21**: 370–383.

4 Avriel M and Penn M (1993). Exact and approximate solutions of the container ship stowage problem. *Comput Indust Engng* **25**: 271–274.

5 Avriel M, Penn M, Shipier N and Witteboon S (1998). Stowage planning for container ships to reduce the number of shifts. *Annal Opns Res* **76**: 55–71.

6 Wilson ID and Roach PA (1999). Principles of combinatorial optimization applied to container-ship stowage planning. *J Heuristics* **5**: 403–418.

7 Wilson ID and Roach PA (2000). Container stowage planning: a methodology for generating computerised solutions. *J Opl Res Soc* **51**: 1248–1255.

8 Saginaw DJ and Perakis AN (1989). A decision support system for container ship stowage planning. *Marine Technol* **26**: 47–61.

9 Shin JY and Nam KC (1995). Intelligent decision support system for container ship auto-stowage planning. *J Korean Inst Port Res* **9**: 19–32.

10 Adlakha V and Kowalski K (1999). On the fixed-charge transportation problem. *OMEGA* **27**: 381–388.

11 Palekar US, Karwan MH, and Zionts S (1990). A branch and bound method for fixed charge transportation problem. *Mngt Sci* **36**: 1092–1105.

12 Gilbert KC and Madan MS (1991). A heuristic for a class of production planning and scheduling problems. *IIE Trans* **23**: 282–289.

13 Hillier FS and Lieberman GJ (1995). *Introduction to Operations Research.* McGraw-Hill, Singapore.

14 Murty KG (1995). *Operations Research: Deterministic Optimization Models.* Prentice Hall, Englewood Cliffs, NJ.

Received July 2000;
accepted November 2001 after one revision