OR Spectrum (2004) 26: 93–116
DOI: 10.1007/s00291-003-0148-0

OR Spectrum

A beam search algorithm for the load sequencing of outbound containers in port container terminals*

Kap Hwan Kim[1], **Jin Soo Kang**[1], **and Kwang Ryel Ryu**[2]

[1] Department of Industrial Engineering , Pusan National University, Changjeon-dong, Kumjeong-ku, Pusan 609-735, South Korea (e-mail: kapkim@pusan.ac.kr)
[2] Department of Computer Engineering, Pusan National University, Pusan, South Korea (e-mail: krryu@pusan.ac.kr)

Abstract. A beam search algorithm was applied to solve the load-sequencing problem in port container terminals. The algorithm was used to maximize the operational efficiency of transfer cranes and quay cranes (QCs) while satisfying various constraints on stacking containers onto vessels. The load-sequencing problem consisted of two decision-making subproblems. In the first subproblem, a pickup schedule was constructed in which the travel route of a transfer crane (TC) as well as the number of containers it must pick up at each yard-bay are determined. In the second subproblem, the load sequence for individual containers was determined. This study suggested a search scheme in which an algorithm to solve the second subproblem is imbedded into the algorithm for the first subproblem. Numerical experiments using practical data were performed to test the performance of the developed algorithm.

Keywords: Load sequencing – Container terminal – Beam search

1 Introduction

The container handling system considered in this study consists of QCs to load (unload) onto (from) containerships, TCs for transferring containers within a marshaling yard, and yard trucks (YTs) for delivering containers between the marshaling yard and QCs. A container terminal yard is divided into multiple blocks (see Fig. 1). A block consists of 20 to 30 yard-bays, each of which usually has four tiers and six stacks. To load a container in a yard onto a ship, a TC moves to a target

* This study was supported in part by the Korea Science and Engineering Foundation through the Center for Intelligent and Integrated Port Management Systems (CIIPMS) at Dong-A University. Authors also appreciate Director In-Tae Baek in Korea Logistics Network Co. for his help during the data collection.
Correspondence to: K. H. Kim

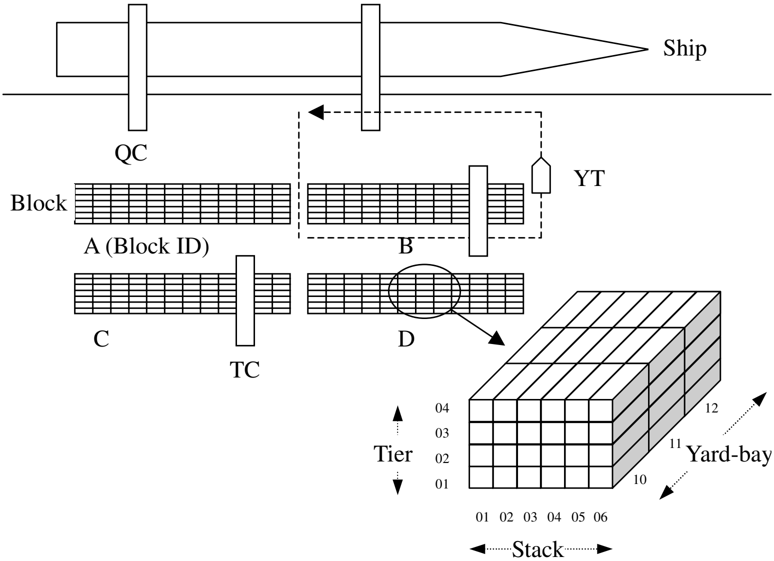

Fig. 1. An overview of container terminal

yard-bay, then its hoist picks up a selected container and takes it to one side of the block and loads it onto a waiting YT. A TC picks up containers in an order specified by a load planning process. Next, the YT transports the container to a QC. Finally, the QC picks up the container and loads it onto a ship. TCs usually move along the yard-blocks that are laid out parallel to the berth, while QCs can move on the rail that also runs parallel to the berth. A TC or a QC cannot pass another TC or QC.

Figure 2 shows a containership that has 28 ship-bays each of which consists of many stacks in hold and on deck. Hatch covers separates stacks on deck from those in hold. Thus, for each ship-bay, containers for slots in hold must be completely loaded before the loading operation into slots on deck can begin.

The ship operation consumes a large portion of the turnaround time of containerships in ports. The ship operation of a container ship consists of unloading inbound containers and loading outbound containers. Because inbound containers are usually unloaded onto a designated open space and there are fewer requirements to be satisfied in case of the unloading operation than in case of the loading operation, the sequencing of unloading operations is relatively easier than that of loading operations. In the loading operation, containers to be loaded into slots of a ship must satisfy various constraints on the slots pre-specified by a stowage planner. Also, locations of outbound containers may be scattered over a wide area in a marshaling yard. The time required for loading operations depends on the cycle time of QCs and TCs. Also, the cycle time of a QC depends on the loading sequence of slots, while the cycle time of a TC is affected by the loading sequence of containers in the yard. This study assumed that a transfer quay crane is exclusively assigned to each quay crane during the ship operation.

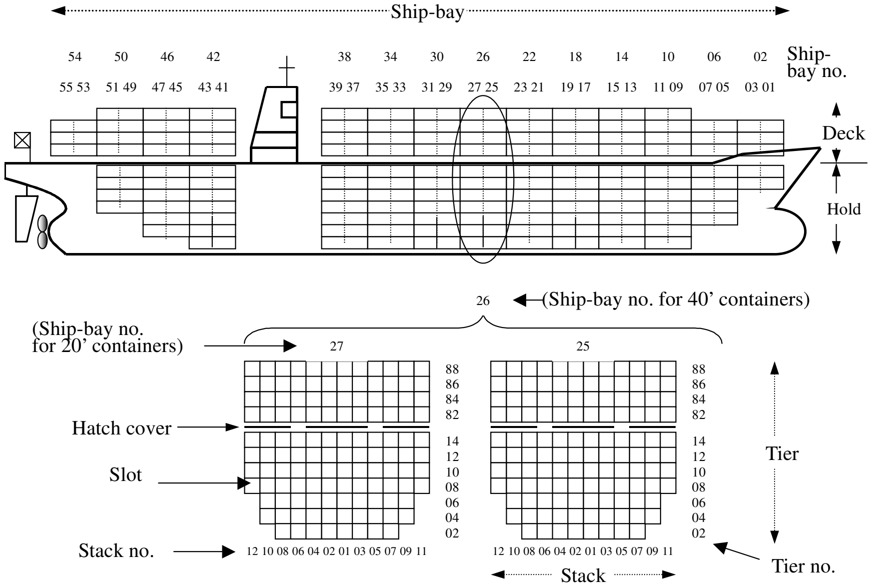

Fig. 2. Cross-sectional view of a containership

Research on load sequencing can be classified into three types according to its problem-solving approach: mathematical programming approaches (Cho [2], Kim [6]), heuristic algorithms (Beliech [1], Cojeen [3], Gifford [4]), and meta-heuristic approaches (Kim [5], Kozan [7], Ryu [10]). Research can also be classified by the scope of the problem. Some research has addressed the pickup scheduling problem in which the travel route of each yard crane and the number of containers to be picked up at each yard-bay on the route are determined during the loading process of a vessel (Kim [5], Kim [6], Narasimhan [7], Ryu [10]). Ryu et al. [10] suggested an algorithm based on "the ant system" for solving the pickup scheduling problem for TCs, which is a sub-problem of the load-sequencing problem. The performance of their algorithm will be compared with that of the algorithm in this study. Other research has attempted to determine the loading sequence of individual containers in the marshaling yard and slots in the vessel, a process that requires more detailed scheduling than does the pickup scheduling (Beliech [1], Cho [2], Cojeen [3], Gifford [4], Kozan [7]).

This study is different from previous studies in the following three aspects:

1. Many practical constraints and objective functions of the load-sequencing problem are considered in the algorithm. Examples are the travel distance of TCs, the handling convenience of TCs and QCs, the maximum height of a stack in hold, the maximum total weight of containers on a hatch cover, and the conformity of weights of loaded containers to the weight class specified in a stowage plan.
2. The loading sequence of slots in a vessel and containers in a marshaling yard are simultaneously determined.

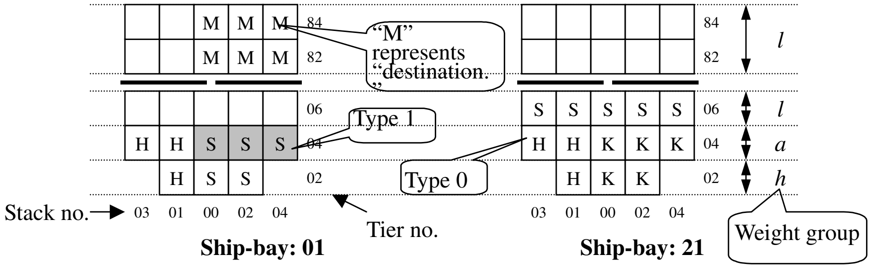

Fig. 3. An illustration of a stowage plan

3. Instead of simple rules, a meta-heuristic searching algorithm, called the filtered beam search, is used to obtain a solution. Although there has been some research that attempted to simultaneously determine the sequence of slots and containers, simple heuristic rules based on planners' intuition have been usually applied.

2 Problem definition of the load sequencing

By using stowage plans such as the one shown in Figure 3, shipping companies specify the port of destination (H, M, S, K), the size (20', 40', or 45'), the type (dry full container, refrigerated containers, empty containers, containers with dangerous cargo, etc.) and the weight group (light (l), medium (a), heavy (h)) of the container allowed to be loaded into each slot of a vessel. Containers of the same size and type and bound for the same destination is said to be in the same class. Because ship-bay numbers in Figure 3 are odd, all the outbound containers to be loaded into the ship-bays are 20' containers. Also, before the load sequencing for a vessel begins, planners usually construct a work schedule of QCs, which shows the sequence of ship-bays that each quay crane should perform discharging and loading operations, for the vessel, as shown in Figure 4. Then, load planners determine the loading sequence of slots in the vessel and containers in the yard. In the process, load planners use the yard map such as the one in Figure 5, which shows the distribution of containers in the yard. The yard map also shows the destination, the weight group, and the type of each container stacked in each position in the yard. An illustrative example of a load sequence list is provided in Table 1, which shows the load sequence of containers, the locations of the containers in the yard before loading, and their locations in the vessel after loading.

The following summarizes what load planners must consider during the load sequencing process. Some considerations are related to the operation of QCs, while others are related to the operation of TCs. Because many requirements must be satisfied, the load sequencing process is very time-consuming for planners and requires intensive computer support. In the load-sequencing algorithm in this study, some of the considerations are treated as constraints – which it was attempted to satisfy by imbedding them into the search procedure, while others were treated as factors in the objective function, as follows:

Work schedule for QC 1

Ship no: AA. Voyage no.: 05 Date: June 25, 2002

Sequence	Ship-bay no.	Hold/deck	Unload/load	No. of containers		
				20ft	40ft	45ft
1	01	H	L	8		
2	01	D	L	6		
3	03	D	L	14		
4	05	H	L	27		
5	05	D	L	3		

Fig. 4. An illustration of a work schedule for QC 1

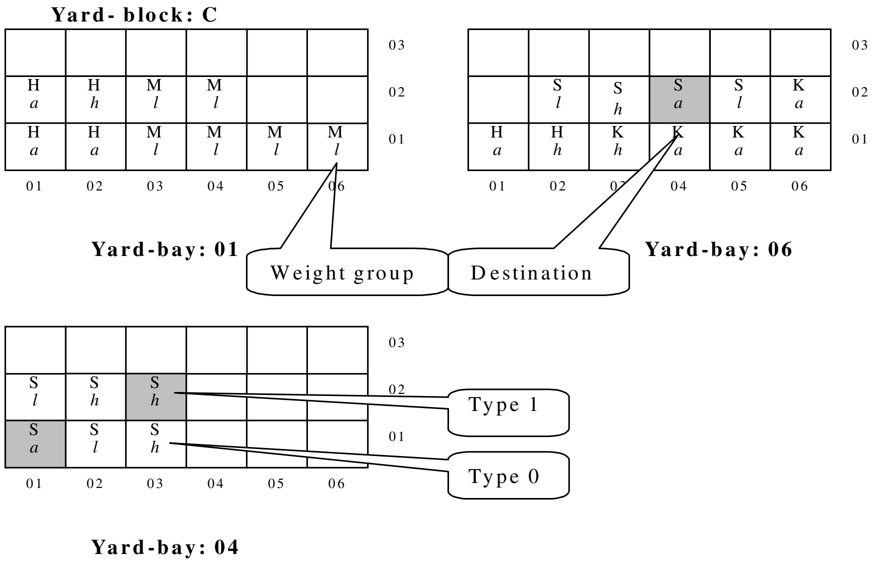

Fig. 5. An illustration of a yard map

Table 1. An example of a load sequence list

QC no.	Sequence	Container number	Location in yard	Location in vessel
1	1	MFU8408374	C-06-03-02*	01-00-02**
1	2	DMU2975379	C-06-02-02	01-02-02
1	3	DMU2979970	C-06-01-02	01-01-02
1	4	OLU0071308	C-01-02-02	01-01-04
1	5	MTU4015162	C-01-01-02	01-03-04
…	…	…	…	…

* Block no. – yard-bay no. – stack no. – tier no., ** Ship-bay no. – stack no. – tier no.

Objectives related to operation of QCs

1.1 First fill slots in the same stack in the hold. Filling slots in the same hold consecutively speeds to the loading operation of QC, because, in that case, automatic positioning function may used to move the spreader of QC.
1.2 First stack containers onto the same tier on deck. As objective 1.1, stacking containers onto the same tier on deck consecutively speeds up the lashing operation. Because preferences of QC operators are considered in objectives 1.1 and 1.2 and the preferences may be different from a terminal to another, the objectives may be modified if preferences of QC operators in a terminal are different from 1.1 and 1.2.
1.3 Stack containers of weights included in the same weight group as specified in the stowage plan.

Objectives related to operation of TCs

2.1 Minimize the travel time of TCs.
2.2 Minimize the number of rehandles.
2.3 Pick up containers in locations nearer to the transfer point earlier than those located farther from the transfer point.

Constraints related to operation of QCs

3.1 Follow precedence relationships among slots due to work schedules for QCs and due to relative positions between slots in a ship-bay.
3.2 Do not violate the maximum allowed total weight of the stack on deck.
3.3 Do not violate the maximum allowed height of the stack of a hold.
3.4 Load the same class, which is defined by the destination port, the size, and the type, of containers as specified in the stowage plan.

Constraints related to operation of TCs

4.1 Maintain the distance between adjacent TCs by at least 5 yard-bays.

For the problem formulation, the following notations are introduced:

Indices

i, j = Indices for containers in the yard.
s, t = Indices for slots in the vessel.
k = Index for load sequences for QCs.
c = Index for QCs.
u = Index for stacks in the vessel.

Problem data

m	=	The number of QCs.
n_c	=	The number of containers (slots) to be loaded (filled) by QC c.
n	=	The total number of containers to be loaded into the vessel. $n = \sum_{c=1}^{m} n_c$.
c_w	=	Penalty for the difference between the weight group of a container assigned to a slot and the weigh group planned for the slot in the stowage plan.
c_d	=	Penalty for the inconvenience of the loading operation for slots on deck by QCs. This is due to changing tiers on deck during the loading operation.
c_h	=	Penalty for the inconvenience of the loading operation for slots in hold by QCs. This due to changing stacks in hold during the loading operation.
a_t	=	Penalty for unit travel time by TCs.
a_r	=	Penalty of re-handling a container by TCs.
a_h	=	Penalty for the inconvenience of the transfer operation by TCs. This penalty is applied when a container is picked up before a container – which is located nearer to the transfer point than the container – is picked up.
g_s^o	=	Planned weight group – which is specified in the stowage plan – for slot s.
g_i	=	Weight group of container i.
w_i	=	Weight of container i.
h_i	=	Height of container i.
α_{st}	=	1, if slot s and t are located in the same tier on deck; 0, otherwise. $a_{ss} = 0$.
β_{st}	=	1, if slot s and t are located in the same stack in hold; 0, otherwise. $\beta_{ss} = 0$.
λ_{st}	=	1, if slot s and t are located in the same stack on deck; 0, otherwise. $\lambda_{ss} = 0$.
t_{ij}	=	Travel time of TCs from the location of container i to the location of container j.
t_i	=	Transfer time of container i by a TC for picking up and putting down it on a YT.
γ_{ij}	=	1, if container i and j are in the same stack of the yard and container i is located below container j; 0, otherwise.
δ_{ij}	=	1, if container i is located farther from the transfer point than container j in a yard-bay; 0, otherwise.
θ_{is}	=	1, if the class of container i is the same as the container class of slot s specified in the stowage plan; 0, otherwise.
w_u^m	=	The maximum allowed total weight of stack u on deck.
h_u^m	=	The maximum allowed height of stack u in hold.
M	=	A very large positive number.

Sets of indices

P	=	The set of pairs of slots with a precedence relationship between slots due to relative positions in a ship-bay or due to the work schedule specified for each QC. If $(s, t) \in P$, then slot s must be filled before slot t is filled with a container.
W_c	=	The set of slots assigned to QC c in the work schedule.
W_c^D	=	The set of slots on deck among slots in W_c.
W_c^H	=	The set of slots in hold among slots in W_c.
V_u	=	The set of slots in stack u.
T^D	=	The set of stacks on deck.
T^H	=	The set of stacks in hold.
U	=	The set of pairs of containers that cannot be transferred by TCs at the same time because of interferences between TCs.

Decision variables

X_{isk}^c	=	1, if container i is picked up in the k^{th} order and stacked into slot s in the vessel by QC c; 0, otherwise.
S_i	=	The transfer starting time for container i by a TC.
T_i	=	The transfer completion time for container i by a TC.
Z_{ij}	=	1, if the transfer of container i by a TC is completed before starting the transfer of container j; 0, otherwise.

Then, the load-sequencing problem can be formulated as follows:

$$\begin{aligned} Min \quad & \left(c_d \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{c=1}^{m} \sum_{s \in W_c^D} \sum_{t \in W_c^D} \sum_{k=1}^{n_c - 1} (1 - \alpha_{st}) X_{isk}^c X_{jt(k+1)}^c \right. \\ & + c_h \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{c=1}^{m} \sum_{s \in W_c^H} \sum_{t \in W_c^H} \sum_{k=1}^{n_c - 1} (1 - \beta_{st}) X_{isk}^c X_{jt(k+1)}^c \\ & + c_w \sum_{i=1}^{n} \sum_{c=1}^{m} \sum_{s \in W_c} \sum_{k=1}^{n_c} |g_s^o - g_i| X_{isk}^c \\ & + a_t \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{c=1}^{m} \sum_{s \in W_c} \sum_{t \in W_c} \sum_{k=1}^{n_c - 1} t_{ij} X_{isk}^c X_{jt(k+1)}^c \\ & \left. + \ a_r \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{ij} Z_{ij} + a_h \sum_{i=1}^{n} \sum_{j=1}^{n} \delta_{ij} Z_{ij} \right) \end{aligned} \tag{1}$$

subject to

$$\sum_{i=1}^{n} \sum_{s \in W_c} X_{isk}^c = 1 \text{ for } c = 1, 2, \ldots, m \text{ and } k = 1, 2, \ldots, n_c, \tag{2}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{n_c} X_{isk}^c = 1 \text{ for } c = 1, 2, \ldots, m \text{ and } s \in W_c, \tag{3}$$

$$\sum_{c=1}^{m}\sum_{s\in W_c}\sum_{k=1}^{n_c} X_{isk}^c = 1 \text{ for } i = 1, 2, \ldots, n, \tag{4}$$

$$\sum_{p=1}^{n}\sum_{q=1}^{n}\sum_{s\in W_c}\sum_{t\in W_c}\sum_{r=1}^{k-1}(t_p + t_{pq})X_{psr}^c X_{qt(r+1)}^c - S_i \le M(1 - X_{iuk}^c)$$
$$\text{for } i = 1, 2, \ldots, n, c = 1, 2, \ldots, m, u \in W_c, k = 1, 2, \ldots, n_c, \tag{5}$$

$$S_i + t_i = T_i \text{ for } i = 1, 2, \ldots, n, \tag{6}$$

$$S_j - T_i \le MZ_{ij} \text{ for } i, j = 1, 2, \ldots, n, \tag{7}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{p} X_{isk}^c - \sum_{i=1}^{n}\sum_{k=1}^{p} X_{itk}^c \ge 0 \text{ for } c = 1, 2, \ldots, m, \ \text{all}(s, t) \in P,$$
$$\text{and } p = 1, 2, \ldots, n_c, \tag{8}$$

$$\sum_{i=1}^{n}\sum_{c=1}^{m}\sum_{s\in V_p}\sum_{k=1}^{n_c} w_i X_{isk}^c \le w_p^m \text{ for all } p \in T^D \tag{9}$$

$$\sum_{i=1}^{n}\sum_{c=1}^{m}\sum_{s\in V_p}\sum_{k=1}^{n_c} h_i X_{isk}^c \le h_p^m \text{ for all } p \in T^H, \tag{10}$$

$$X_{isk}^c \le \theta_{is} \text{ for } i = 1, 2, \ldots, n, c = 1, 2, \ldots, m,$$
$$s \in W_c, k = 1, 2, \ldots, n_c, \tag{11}$$

$$Z_{ij} + Z_{ji} = 1 \text{ for all } (i, j) \in U \tag{12}$$

$$X_{isk}^c = 0 \text{ or } 1 \text{ for } i = 1, 2, \ldots, n, c =, 1, 2, \ldots, m,$$
$$s \in W_c, k = 1, 2, \ldots, n_c, \tag{13}$$

$$Z_{ij} = 0 \text{ or } 1 \text{ for } i, j = 1, 2, \ldots, n, \tag{14}$$

$$S_i, T_i > 0. \tag{15}$$

The terms of (1) correspond to objectives, 1.1, 1.2, 1.3, 2.1, 2.2, and 2.3, respectively. Each place in the loading sequence, each container, and each slot are assigned the value of one once and only once in the feasible solution by constraints (2), (3), and (4). Constraints (5), (6) and, (7) define variables S_i, T_i, and Z_{ij}. Constraints (8), (9), (10), (11), and (12) correspond to constraints 3.1, 3.2, 3.3, 3.4, and 4.1. Precisely defining, the value of t_i usually depends on the sequence of transfer, but, for the simplicity of the formulation, this study assumed that t_i is independent of the transfer sequence and has a constant value.

The objective function has quadratic terms as well as linear terms, and some decision variables are 0-1 binary variables. Considering loading containers numbers up to higher than 1000, developing a heuristic algorithm for near optimal solutions is a practical approach. Thus, a heuristic algorithm is proposed in the next section.

Table 2. Constructed yard-clusters

Cluster-ID	Port of destination	Size (ft)	Type	Stack location	Number of containers
H201*	H	20	0	C-01	4
M201	M	20	0	C-01	6
S204	S	20	0	C-04	4
S214	S	20	1	C-04	2
H206	H	20	0	C-06	2
S206	S	20	0	C-06	3
S216	S	20	1	C-06	1
K206	K	20	0	C-06	5

* H201 : H (port of destination), 2 (20ft), 0 (type), 1 (location).

3 A beam search algorithm for load-sequencing

This section introduces a beam search algorithm for the load-sequencing problem. The beam search method is similar to the branch and bound method in that both methods reject unpromising nodes in a large search tree, and thus save time and effort to search for the branches of the search tree growing from the rejected nodes. The filtered beam search does this by using a total cost evaluation function (a cost estimate projected from the current partial solution to a complete solution) and one-step priority evaluation function (a cost estimate only to the next step chosen by a simple priority rule). The calculation of a total cost evaluation function usually takes much longer than that of a one-step priority evaluation function. Thus, the filtered beam search first selects candidate solutions (called filtered nodes), whose number is the same as the filter-width, of the next stage by using a one-step priority evaluation function. And then, the filtered beam search procedure evaluates a total cost evaluation function of each filtered node and selects beam nodes, whose number is the same as the beam-width, among filtered nodes (Ow and Morton [8]).

To apply the beam search algorithm, first, a list of yard-clusters of containers is constructed. A yard-cluster is defined as a collection of containers of the same size and type (dry container, refrigerated container, empty container, container with dangerous cargo, etc.) that have the same destination port and which are stacked in the same yard-bay. Considering the example in Figure 6, the list of yard-clusters can be as shown in Table 2.

Two types of beam search are used to search for solutions. The load-sequence of yard-clusters is determined by the first search algorithm, which is called the filtered beam search. The load-sequence of individual containers is determined by the second beam search. The first beam search procedure starts from constructing initial beam nodes. For each initial beam node, nodes in the next stage are generated and filtered by using a total cost evaluation function to select a beam node. For the selected beam node, the sequence of individual containers is determined by the second beam search procedure. The second beam search procedure follows the normal beam search procedure in which, at each stage, beam nodes of the next stage are selected by one-step evaluation function.

Sequencing yard-clusters is equivalent to constructing the pickup schedule (Kim [5], Kim [6], Narasimhan [7], Ryu [10]) that specifies the visiting sequence of yard-bays and the number of containers to pick up at each visiting yard-bay. To determine the sequence of yard-clusters, the first search algorithm needs work schedules of QCs, a list of yard-clusters, and a stowage plan for vessels. The first search algorithm attempts to minimize the total travel time of TCs and to satisfy constraints related to the sequence of yard-clusters. That is, the first search algorithm solves the problem with objective 2.1 and constraints 3.1, 3.4, and 4.1 in the previous section.

The second beam search determines the sequence of individual containers to maximize the handling convenience of QCs and TCs and the degree of satisfaction of the weight requirement. The second beam search also attempts to find solutions to satisfy constraints on the maximum weight of a stack on deck and the maximum height of a stack in hold. Once again, the loading sequence of individual slots must obey the precedence relationships among slots (e.g.,the rule that slots in the bottom must be filled before slots on the top are filled with containers). Thus, the second search procedure solves the problem with objectives 1.1, 1.2, 1.3, 2.2, and 2.3 in the previous section and constraints 3.1, 3.2, and 3.3.

One approach to solving the load-sequencing problem is to sequentially solve two subproblems: sequencing yard-clusters and then sequencing individual containers for the resulting sequence of clusters. One of the difficulties of the sequential approach is that, because the constraints of the second subproblem are not considered when solving the first subproblem, the final solution of the first subproblem may result in an infeasible solution to the second subproblem. Therefore, the two subproblems must be solved simultaneously. Thus, in this study, the search procedure for the second subproblem (the second beam search) is imbedded within the search procedure for the first subproblem (the first beam search). During the second beam search for sequencing individual containers in a yard cluster which corresponds to a filtered node selected in the first beam search procedure, if no feasible sequence can be found, then the filtered node is removed from the set of filtered node of the first beam search tree. And then, the second beam search procedure starts again from the next-best filtered node in the first beam search tree.

For the description of the search algorithm, the following notations are introduced:

C	$=$	The set of yard-clusters.
N_e^o	$=$	The e^{th} initial beam node in the first beam search procedure.
b	$=$	The number of initial beam nodes.
N_e	$=$	The current beam node connected to N_e^o.
F	$=$	The set of filtered beam nodes in the first beam search procedure.
f	$=$	The filtered beam width in the first beam search procedure.
$x(N_e)$	$=$	The partial solution – which can be represented by a sequence of yard-cluster for each QC – corresponding to the path from the root node to N_e in the first beam search tree.
$t(x(N_e))$	$=$	The total travel time of $x(N_e)$.

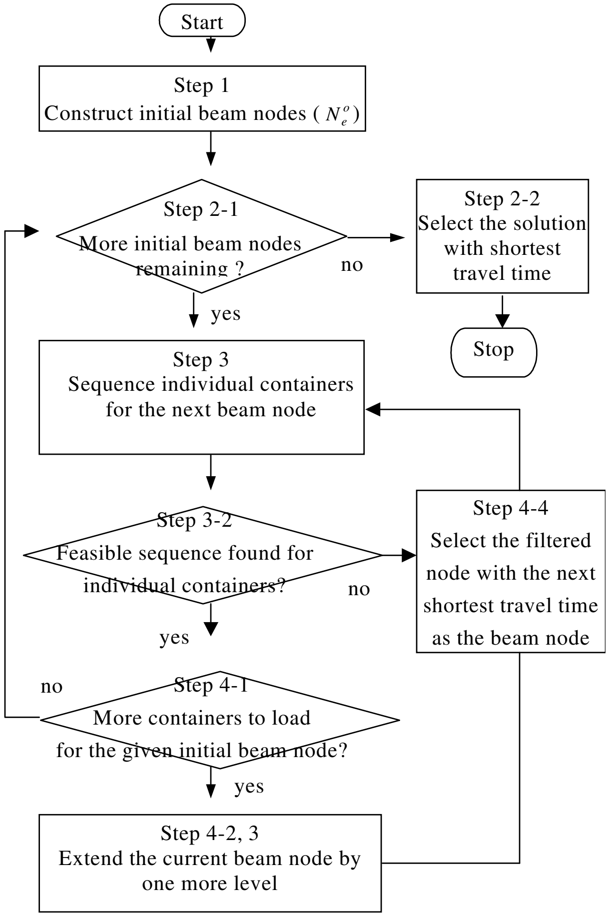

Fig. 6. The overall search procedure for load-sequencing

The overall search procedure is suggested in Figure 6, and a detailed explanation is provided in the following discussion. Note that the main flow of Figure 6 is related to the first subproblem. The beam search procedure for the second subproblem corresponds to Step 3 in Figure 6, which will be described in more detail in Figure 7.

Step 1. (Construct initial beam nodes)

In this step, initial beam nodes are constructed.

Step 1-1. (List candidate yard-clusters for each QC)

For each QC, list all the yard-clusters, from C, which have at least one container that satisfies constraints 3.1 and 3.4. That is, each listed yard-cluster must have

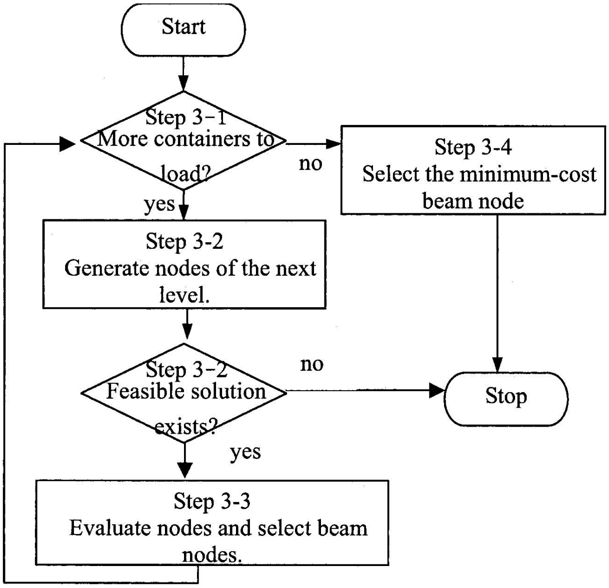

Fig. 7. The second beam search procedure for sequencing individual containers (Step 3)

at least one container that can be loaded into the lowest tier of stacks of the first ship-bay in the work schedule for the QC. Let C_c be the set of yard-clusters listed for QC c.

Step 1-2. (Globally evaluate candidate yard-clusters for each QC)

For each $v \in \mathrm{C}_c$, by a neighborhood search, construct a complete pickup schedule which specifies the visiting sequence of yard-bays and the number of containers to pick up at each visiting yard-bay, for all the slots to be filled in the current hold or deck. The neighborhood search procedure constructs the complete pickup schedule by sequentially selecting the nearest yard-cluster, starting from yard-cluster v, and loading containers, satisfying constraints 3.1 and 3.4 in each selected yard-cluster, as many as possible. The complete pickup schedule for each $v \in \mathrm{C}_c$ is evaluated by the travel time of the TC.

Step 1-3. (Construct and select the initial beam nodes)

List all the possible combinations of m elements – one element from each $C_c, c = 1, 2, \ldots, m$. Delete combinations that violate constraints 4.1. For each remaining combinations, sum all the travel times of m pickup schedules (of Step 1-2) corresponding to m elements (yard-clusters) in the combination. Among all the combinations, b combinations with the shortest total travel times are selected as initial beam nodes $(N_e^o, e = 1, 2, \ldots, b).e = 0$.

Step 2. (Check for the existence of remaining initial beam nodes)

$e = e + 1$. If $e > b$, then select $x(N_t), t = 1, 2, \ldots, b$, with the minimum $t(x(N_t))$ (the total travel time) as the final solution and stop the procedure (Step 2-2). Otherwise, $N_e = N_e^o$ and go to Step 3.

Step 3. (Sequence individual containers)

In this step, for yard clusters corresponding to N_e, individual containers are sequenced, and the sequenced containers are removed from the yard map. If no sequence of individual containers can be constructed without violating constraints 3.1, 3.2 and 3.3, then go to Step 2 when the current level of the first beam search is 1 or go to Step 4-5 when it is greater than 1. If a sequence of individual containers can be constructed, then go to Step 4. Step 3 will be described in more detail later.

Step 4. (Extend the current beam node, N_e, by one level)

Step 4-1. (Check for the existence of remaining containers)

For the current beam node, N_e, if more containers to be loaded exist, go to Step 4-2. Otherwise, go to Step 2.

Step 4-2. (List candidate yard-clusters)

Based on $x(N_e)$, select a QC, among the QCs that have remaining containers to load, which completed the previous work the earliest. Let the selected QC be QC c. Construct C_c by using candidate yard clusters for QC c as in Step 1-1.

Step 4-3. (Select filtered nodes)

To construct F, select f elements (yard-clusters) from C_c with the shortest travel time from the last location of the TC in $x(N_e)$ to the locations of the candidate yard-clusters. This evaluation process is called a "local evaluation." The selected f elements in C_c are called "filtered nodes."

Step 4-4. (Perform the global evaluation)

Perform a global evaluation for all the filtered nodes by the same procedure as the one in Step 1-2.

Step 4-5. (Select a beam node from filtered nodes)

Select the filtered node with the shortest travel time from F. Delete the selected node from F. The selected node becomes the new beam node, N_e. Go to Step 3.

The following describes how to sequence individual containers in Step 3. If Step 3 is performed during level 1 of the first beam search procedure, then the following procedure will be repeated as many times as the number of QCs, while, otherwise, the following procedure is performed once.

Before beginning Step 3, a yard-cluster has already been determined for sequencing individual containers for a QC. In the search tree for sequencing individual containers, one container for loading is determined at each level. Thus, the depth of the search tree is the same as the maximum number of containers that can be transferred from the current yard cluster.

For a more detailed explanation of Step 3, the following notations are used:

d	$=$	The beam width for the second beam search procedure.
r	$=$	Index representing the search level for the second beam search procedure.
B_r	$=$	The set of beam nodes at level r for the second beam procedure.
G_r	$=$	The set of all the generated nodes in level r.
M_r	$=$	A beam node in B_r.
$q(M_r)$	$=$	The total penalty of objectives 1.1, 1.2, 1.3, 2.2, and 2.3 from the root node to M_r.
z	$=$	The number of containers to be sequenced.

Figure 7 shows the overall procedure of Step 3 that is also explained in the following:

$r = 0$.

Step 3-1. (Check for the existence of additional containers to load)

If $r = z$, then select, as the final solution, the beam node with the minimum $q(M_s)$, $s = 1, 2, \ldots, d$, and stop. Otherwise, $r = r + 1$ and go to Step 3-2.

Step 3-2. (Generate nodes for the next level)

For each M_{r-1}, generate all the feasible combinations, which satisfy constraints 3.1, 3.2, and 3.3, of the next candidate slots in the ship-bay and the next candidate containers in the yard-cluster. All the generated feasible combinations become elements of G_r. If no combination, which does not violate the constraints, can be found in the current yard-cluster, then the next filtered node with the next shortest travel time in Step 4-4 is selected as a beam node and repeat Step 3 again.

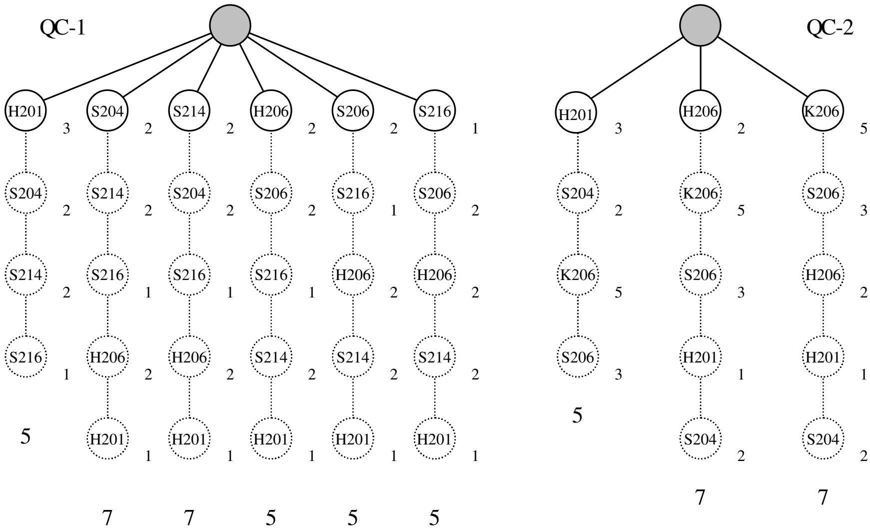

Fig. 8. The result of global evaluation for candidate yard-clusters

Step 3-3. (Evaluate nodes and select beam nodes)

Evaluate $q(y)$ for all the elements $y \in G_r$. Select d nodes with the minimum values of $q(y)$. The selected nodes are included in B_r. Go to Step 3-1.

A numerical example

By using the stowage plan, the yard map, and yard-clusters in Figures 3 and 5 and Table 2, the algorithm in this paper is illustrated in the following.

Step 1-1: Let QC-1 and QC-2 start the loading operation from ship-bay 01 and ship-bay 21 in Figure 3, respectively. $C_1 = \{$H201, S204, S214, H206, S206, S216$\}$ and $C_2 = \{$H201, H206, K206$\}$.

Step 1-2: As shown in Figure 8, to complete all the tasks in the current hold, three 20-foot containers bound for port “H” and five 20-foot containers bound for port “S” must be transferred for QC-1. The numerical value at each node represents the number of containers picked up from the corresponding yard-cluster. The numerical value at the end of each branch represents the travel time required for loading all the containers in hold. For example, let the first yard-cluster be H201. Then, three 20-foot containers can be picked up from H201. Next, the nearest yard-cluster that has 20-foot containers of type 0 bound for port “S” is S204. From yard-cluster S204, two 20-foot containers of type 0 (S204) and two 20-foot containers of type 1 bound for port “S” (S214) are picked up. Because one more 20-foot container of type 1 bound for port “S” is necessary, it is picked up from S216.

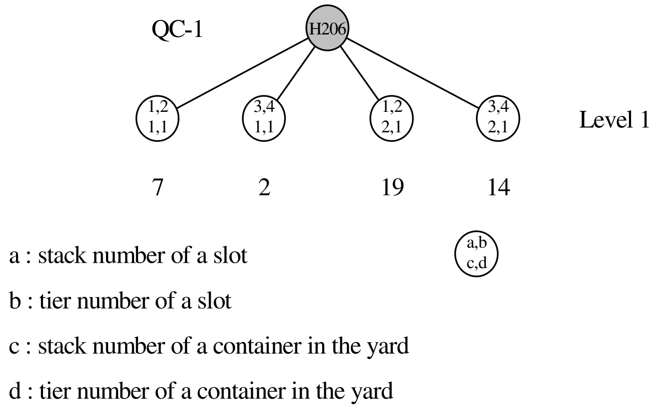

Fig. 9. Constructed nodes of level 1 in the second beam search tree

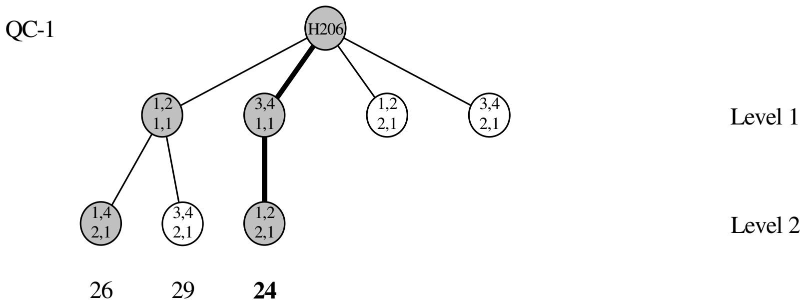

Fig. 10. Nodes constructed to level 2 in the second beam search procedure

Step 1-3: Let b be 5. Then, combinations of $\{$<H206>, (H201)$\}$, $\{$<S206>, (H201)$\}$, $\{$<H201>, (H206)$\}$, $\{$<H201>, (K206)$\}$, and $\{$<S216>, (H201)$\}$ are selected as the initial beam nodes, where the bracket and the parentheses represent the first yard-cluster for QC-1 and QC-2, respectively.

Step 2: $N_1 = N_1^0\{$<H206>, (H201)$\}$.

Step 3-1: Yes, there are containers to be loaded.

Step 3-2: Slots that can be selected as the first slot for QC-1 are (1, 2) and (3, 4), where slots are represented by (stack number, tier number). Containers that can be loaded first are (1, 1) and (2, 1), where containers are also denoted by (stack number, tier number). By combining all the candidate slots and containers, nodes in level 1 (G_1) are constructed as shown in Figure 9.

Step 3-3: Because $d = 2$ in this example, the first two nodes are selected as the beam nodes (elements in B_1). By applying Steps 3-1 through 3-3 once more, the tree as shown in Figure 10 is obtained.

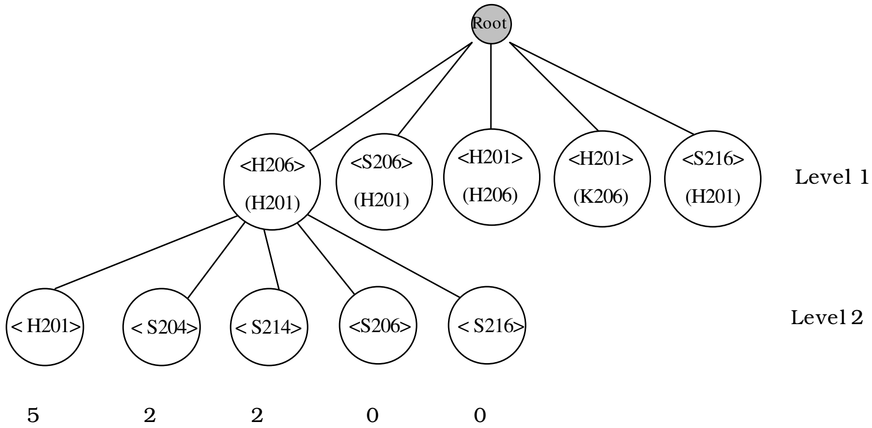

Fig. 11. Local evaluation for filtering nodes constructed at level 2 in the first beam search

Step 3-1: r equals to z. The best solution, which is represented by bold type in Figure 10, is selected as the loading sequence for individual containers for QC-1. The same procedure follows for QC-2 to obtain the loading sequence for individual containers of N_1, {<H206>, (H201)}.

Step 4-1. For N_1, {<H206>, (H201)}, because more containers remain to be loaded, go to Step 4-2.

Step 4-2. The cumulative travel time of the TC for transferring containers of QC-1 is shorter than that of QC-2. Thus, Figure 11 results. C_1 = {<H201>, <S204>, <S214>, <S206>, <S216>}.

Step 4-3: f is set to be 2. Thus, F = {<S206>, <S216>}.

Step 4-4: A global evaluation is performed for the two filtered nodes. It is found that the results are the same as those shown in Figure 12.

Step 4-5: Either of the two filtered nodes can be selected as the next beam node (N_1). This process is repeated until a feasible solution is obtained for the first initial beam node, {<H206>, (H201)}. Then, the algorithm moves to the second initial beam node, {<S206>, (H201)}.

4 Numerical experiments

Two numerical experiments were conducted to test the performance of the beam search algorithm suggested in this paper. The first experiment was conducted to test the sensitivity of algorithm's performance to changes in search parameters b, d, and f. The second experiment was for comparing the performance of the algorithm with two other approaches, the ant system approach (Ryu, 2001), and the neighborhood search algorithm.

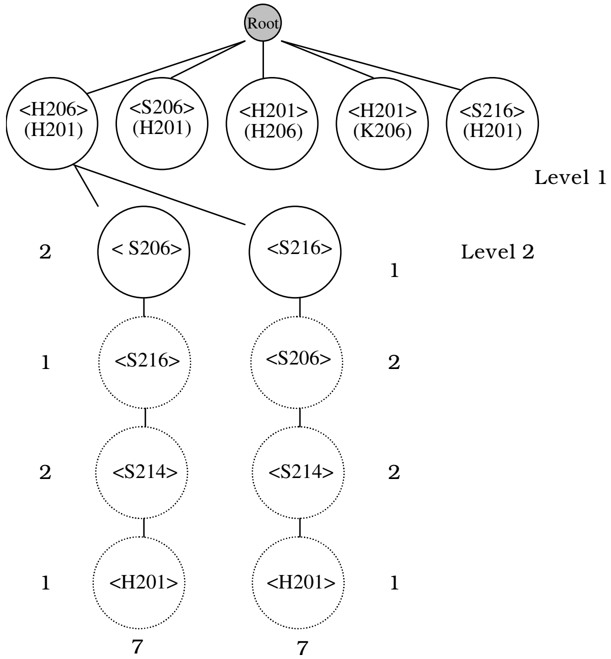

Fig. 12. Global evaluation for nodes filtered at level 2 in the first beam search procedure

The first experiment used a problem with 624 containers, 118 container groups, 32 ship bays, 79 yard-bays, and 4 QCs. Travel time of TCs between adjacent bays, travel time of TCs between different blocks, and travel time of TCs between blocks in different rows were set to be 1, 5, and 25 seconds, respectively. And, the values of the parameters were $c_w = 5$, $c_d = 3$, $c_h = 3$, $a_r = 10$, and $a_t = 1$.

The size of the search space for sequences of yard-clusters depends on the values of b and f, while that for sequences of individual containers is determined by the value of d. Figure 13 shows that the total travel time of TCs is affected significantly by the values of b, while the total travel time is insensitive to the values of f.

Figure 14 shows that both b and f contribute to the reduction of the total weighted penalty in which the penalty of the travel time was included. Figure 15 shows that a larger d results in a smaller total weighted penalty. However, the total travel time of TCs did not change for different values of d, which coincides with our intuition. The computational time was sensitive to the value of b (see Fig. 16) and d (see Fig. 17), while it was insensitive to the values of f.

Note that the algorithm enumerates whole solutions on the sub-tree below one initial beam node and then proceeds to the sub-tree below the next initial beam node. Thus, after the enumeration of the sub-tree below the first initial beam node is completed, the best so far known solution, which is feasible, is obtained. Thus, after then, at any time when the search process is terminated, one or more feasible solutions are available and the best so far feasible solution can be used as the final solution. This is a very useful property of the algorithm in this study. Figures 18 and

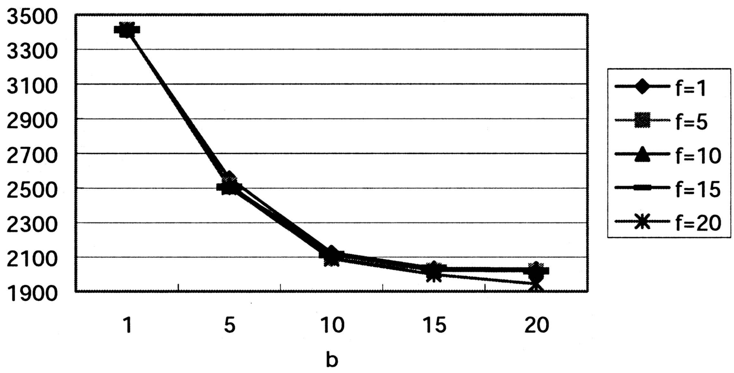

Fig. 13. The total travel time of TCs for different values of b and f ($d = 1$)

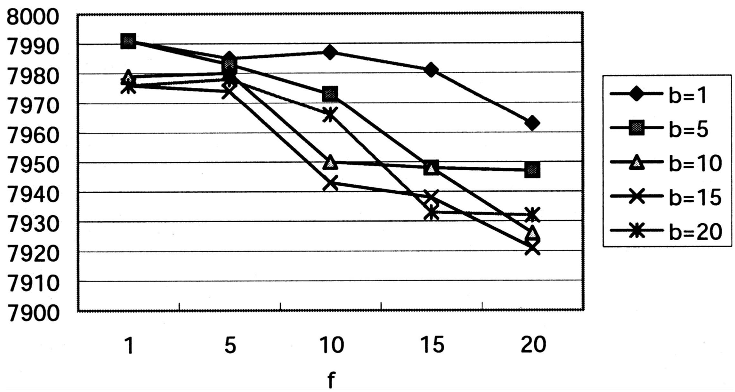

Fig. 14. The total weighted penalty for various values of b and f ($d = 1$)

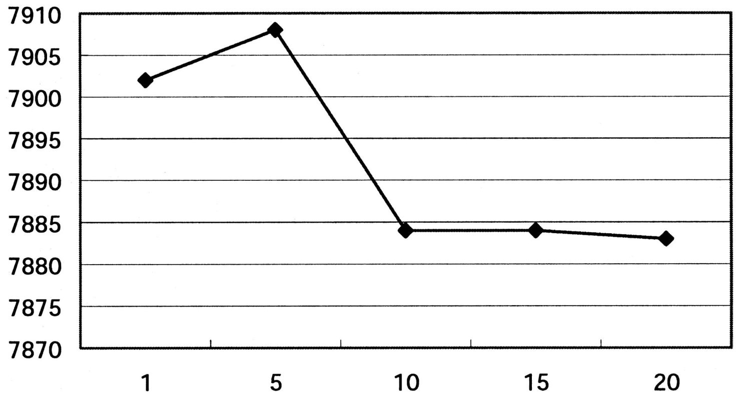

Fig. 15. The total weighted penalty for various values of d ($b = 10$, $f = 15$)

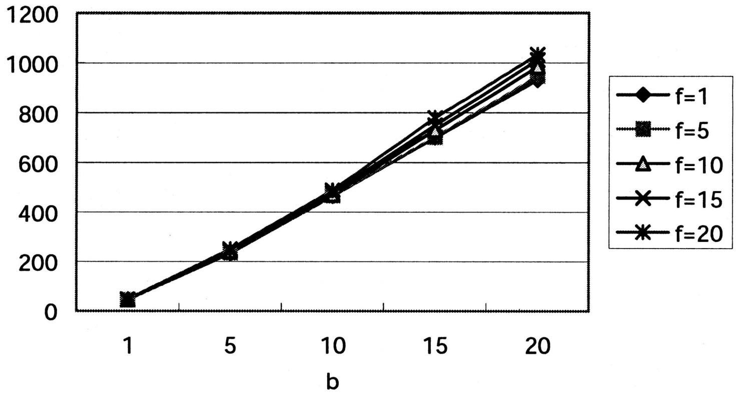

Fig. 16. Computational time (in seconds) for various values of b and f ($d = 1$)

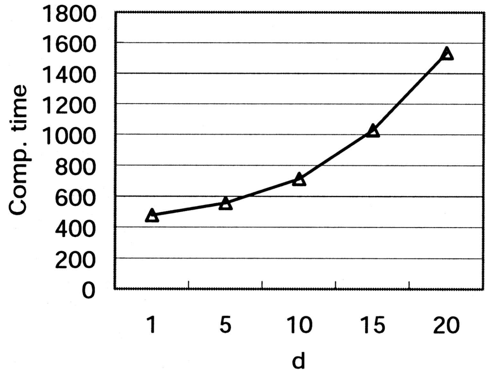

Fig. 17. Computational time (in seconds) for various values of d ($b = 10$, $f = 15$)

19 show how the minimum total travel time of TCs and the minimum total penalty change as the stopping time increases for a sample problem with the number of containers = 624, $b = 30$, $d = 10$, and $f = 15$. That is, these graphs show the trade-off between the quality of the final solution and the computational time.

The second numerical experiment was conducted with six sets of data collected from Pusan Eastern Container Terminal (PECT) in Korea. The size of the problems are listed in Table 3. Problems in Table 3 are representative of real problems in PECT.

Table 4 compares the performance of three solution algorithms: the neighborhood search, the algorithm in this study ($b = 10$, $d = 10$, $f = 15$), and an algorithm based on the ant system (ant algorithm) (Ryu [10]). Note that when values of parameters are set to $b = d = f = 1$, the algorithm in this study reduces to the neighborhood search. In the ant algorithm, the number of repetitions and the number of ants were set to 300 and 600, respectively. A personal computer with Pentium III-600 and 128 Mb-RAM was used for the numerical experiment. The algorithms in this study and in the neighborhood search were programmed by using

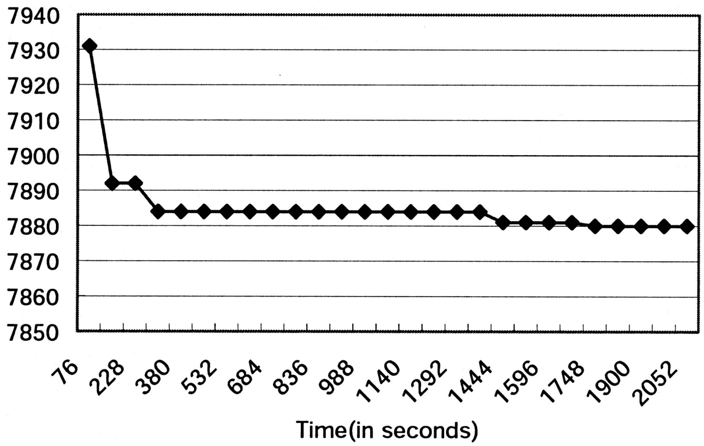

Fig. 18. The change in the minimum total travel time of TCs with respect to the stopping time

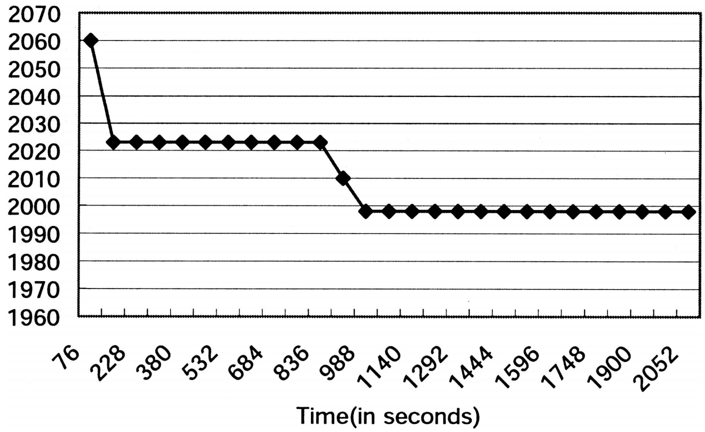

Fig. 19. The change in the minimum total penalty with respect to the stopping time

Table 3. Size of sample problems used in the second experiment

Problem number	Number of containers	Number of container groups	Number of ship bays	Number of yard bays	Number of QCs
1	313	36	18	21	2
2	624	118	32	79	4
3	653	117	21	59	3
4	1012	223	36	130	4
5	1304	242	23	126	3
6	1340	352	43	185	4

Table 4. Performance of three algorithms

	Performance	Neighborhood	Beam search	Ant algorithm
Problem 1	Total travel time	170	168	326
	Total penalty*	1,029	981	1,132
	Comp. time (sec)	13	173	1,410
Problem 2	Total travel time	3,413	2,023	3,280
	Total penalty	7,956	7,892	9,920
	Comp. time	45	712	4,482
Problem 3	Total travel time	1,573	1,467	1,549
	Total penalty	10,324	10,127	10,603
	Comp. time	43	689	4,917
Problem 4	Total travel time	5,866	3,278	5,483
	Total penalty	12,863	12,603	14,232
	Comp. time	74	1,289	10,552
Problem 5	Total travel time	6,671	6,532	6,690
	Total penalty	19,388	17,832	22,954
	Comp. time	115	2,513	12,554
Problem 6	Total travel time	9,295	6,677	8,740
	Total penalty	17,865	17,646	20,787
	Comp. time	111	2,498	11,234

tab4

* The penalty of the travel distance was excluded from the total penalty.

JAVA, while the ant algorithm was programmed by using C++. It is known that the processing speed of C++ is 4 to 5 times faster than that of JAVA.

The beam search algorithm obtained solutions higher in quality than those found by the neighborhood search, but at a cost of higher computational time. Note that the computational time can be adjusted by adjusting the values of b, d, and f, or by specifying the stopping time. The beam search algorithm in this study outperformed the ant algorithm in all three measures of performance. The difference between the two algorithms is due to the fact that, in the ant algorithm, the search procedure is hierarchically divided into two stages: sequencing yard-clusters, and sequencing individual containers in a yard and slots in a vessel. Note that, in this study, the two decisions are integrated and made simultaneously.

5 Conclusion

This paper discusses the load-sequencing problem for outbound containers in port container terminals in which TCs and YTs are used in the marshaling yard. Various constraints and objectives of the load-sequencing problem were introduced. A beam search algorithm was suggested to minimize the handling time of TCs and QCs, and to satisfy various constraints for loading containers. The algorithm in this paper has the following strength: various additional constraints and objectives can be considered without significantly modifying the algorithm, the pickup sequence by TCs in the yard and the loading sequence of slots in the vessel are determined simultaneously, the computational time can be adjusted by users, and the relative importance of elements in the objective function can be modified by users by adjusting the values of parameters of the objective elements.

A sensitivity analysis was performed to show how various performance measures are related to the values of parameters of the beam search algorithm. It was shown that the beam search in this paper outperforms the ant algorithm in the values of objective functions and the computational time.

References

1. Beliech Jr. D E (1974) A proposed method for efficient pre-load planning for containerized cargo ships. Master's Thesis, Naval Postgraduate School, Monterey, California
2. Cho D W (1982) Development of a methodology for containership load planning. Ph.D. Dissertation, Oregon State University
3. Cojeen H P, Dyke P V (1976) The automatic planning and sequencing of containers for containership loading and unloading. In: Pitkin, Roche, Williams (eds) Ship operation automation, pp 415–423. North Holland, Amsterdam
4. Gifford L (1981) A containership load planning heuristic for a transtainer-based container port. Master's Thesis, Oregon State University
5. Kim K H, Kim K Y (1999) Routing straddle carriers for the loading operation of containers using a beam search algorithm. Computers & Industrial Engineering 36: 109–136
6. Kim K H, Kim K Y (1999) An optimal routing algorithm for a transfer crane in port container terminals. Transportation Science 33 (1): 17–33
7. Kozan K, Preston P (1999) Genetic algorithms to schedule container transfers at multimodal terminals. International Transactions in Operational Research 6: 311–329
8. Ow P S, Morton T E (1988) Filtered beam search in scheduling. International Journal for Production Research 26 (1): 35–62.
9. Narasimhan A, Palekar U S (2002) Analysis and algorithms for the transfer routing problem in container port operations. Transportation Science 36 (1): 63–78
10. Ryu K R, Kim K H, Lee Y H, Park Y M (2001) Load sequencing algorithms for container ships by using metaheuristics. Proceedings of 16th International Conference on Production Research (CD-ROM), Prague, Czech Republic