Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers & Operations Research 33 (2006) 940–954

computers & operations research

www.elsevier.com/locate/cor

A heuristic rule for relocating blocks

Kap Hwan Kim*, Gyu-Pyo Hong

Department of Industrial Engineering, Pusan National University, Changjeon-dong, Kumjeong-ku, Busan 609-735, Republic of Korea

Available online 25 September 2004

Abstract

One of the most important objectives of the storage and pickup operations in block stacking systems is to minimize the number of relocations during the pickup operation. This study suggests two methods for determining the locations of relocated blocks. First, a branch-and-bound (B&B) algorithm is suggested. Next, a decision rule is proposed by using an estimator for an expected number of additional relocations for a stack. The performance of the decision rule was compared with that of the B&B algorithm.
© 2004 Elsevier Ltd. All rights reserved.

Keywords: Relocation; Block stacking; Branch and bound algorithm; Heuristic rule

1. Introduction

Block stacking, illustrated in Fig. 1, is a popular stacking method for efficiently utilizing storage space. A stack of blocks consists of multiple blocks that are stacked in the vertical direction, and a bay consists of multiple stacks, as shown in Fig. 1. Fig. 1 also shows the pickup sequence of blocks in a bay. Because blocks are stacked in the vertical direction, relocations must be performed for retrieving a block (we call a "target block") that is not on the top tier. Thus, despite the efficiency in the utilization of space in block stacking, the handling cost resulting from relocations is a serious problem.

There are many practical examples of items for block stacking such as boxes, pallets, marine containers, and steel plates. Marine containers are usually stacked in a container yard before (after) they are loaded (unloaded) into (from) a vessel. In container yards, a bay of containers consists of 6–10 stacks each of

* Corresponding author. Tel.: +82-51-510-2419; fax: +82-51-512-7603.
E-mail address: kapkim@pusan.ac.kr (K.H. Kim).

0305-0548/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2004.08.005

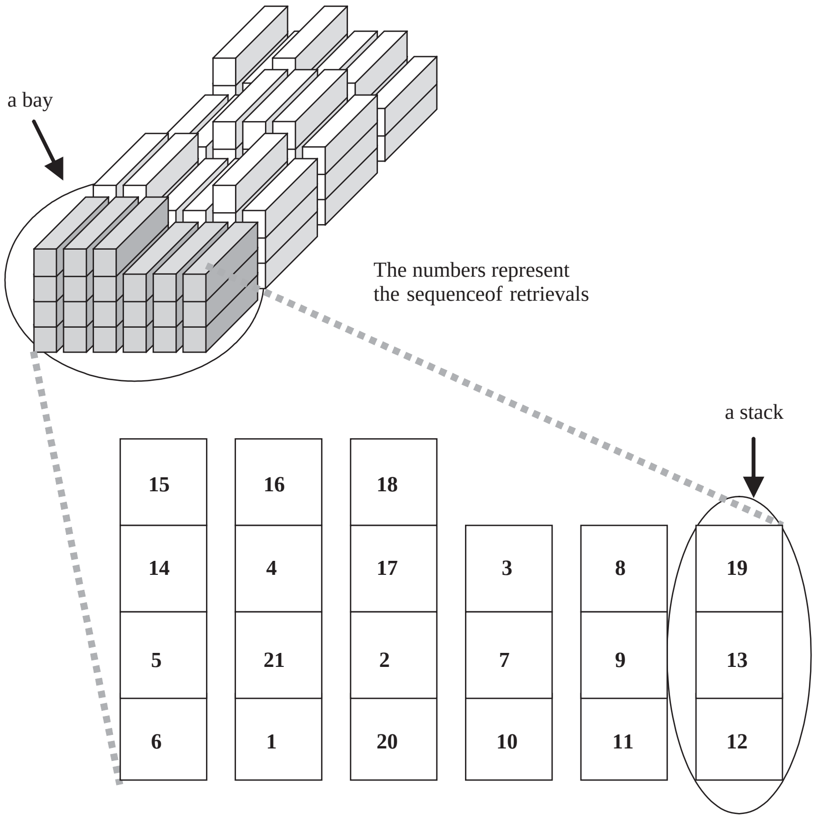

Fig. 1. An illustration of a yard-bay of blocks with a fixed sequence of picks-up.

which consists of 4–7 containers. For steel plates in warehouses, a stack consists of 30–50 plates. It takes about 1–2 min to relocate a marine container or a steel plate by a crane.

The following illustrates a typical example—port container yards—of the process of determining positions of storage locations, pickup sequence, and positions for relocations. The process of determining storage locations of outbound containers can usually be decomposed into three stages: the space allocation stage, the stage of locating individual containers, and the stage of locating relocated containers during pickup operations. In the space allocation stage, the yard-bays, which will be allocated to each vessel for future arrivals of containers, is pre-planned. The decision considers travel distances and congestions of yard trucks and yard cranes during the ship operation. When an outside truck arrives at the gate of container terminals, it is directed to a pre-planned yard-bay for the group of the delivered container. The exact storage slot in the yard-bay for the container is usually determined by the crane operator during the transfer operation. At that time, consideration should be given to the storage location so that heavier containers are located in upper tiers, because heavier containers are likely to be retrieved earlier so that they can be loaded in lower tiers in the vessel. Despite the careful storage of arriving containers, it is possible that heavier containers are located in lower tiers in the yard, then, relocation movements cannot be avoided to pick up heavier containers first. Because storage locations of relocated containers affect the future number relocations during the pickup operations, they also must be determined carefully. For inbound (import) containers, planners have to pre-assign yard-bays before they are unloaded from a vessel. Once yard-bays are allocated, the inbound containers are stacked into the allocated yard space. When an outside truck arrives in a random manner and requests yard cranes to pick up a specific container, the yard cranes transfer it from the yard to the outside truck. Because the outside truck requests a specific container without consideration of its position in the yard-bay, relocations of containers on the top of the requested container must occur frequently. If the terminal operator has any information on the pickup sequence, then the information must be utilized in determining storage locations of unloaded containers or relocated containers so that the number of relocations is minimized.

In summary, for minimizing the number of relocations, first, the storage location of incoming blocks must be efficiently determined. Second, locations of relocated blocks must also be carefully determined. Lastly, the pickup sequence of blocks must be determined in a way that the pickup sequence satisfies constraints on the sequence and that the number of relocations is minimized.

This study addresses the second and the third decision problems. It is assumed that the initial configuration of a yard-bay is given and the pickup priorities among blocks in the yard-bay are given. The objective is to minimize the total number of relocations during the pickup operation of all the blocks in the yard-bay. The constraint is the pickup priorities among blocks in the yard-bay. The following assumptions will be introduced:

1. The precedence of pickups among blocks is known.
2. The possibility of pre-relocations, for reducing the number of future relocations, is excluded. That is, relocations occur only at the moment when a target block is to be picked up.
3. Blocks are relocated to other stacks in the same bay.

Until recently, a limited number of studies have addressed the storage location problem or the retrieval problem considering relocations. Watanabe [1] suggested a simple method, called an accessibility index, to estimate the number of relocations in container terminals for both the straddle carrier system and the transfer crane system. Castilho and Daganzo [2] analyzed import container terminals and proposed various operation strategies for import container yards. They also suggested a formula for estimating the number of relocations. Kim [3] suggested a formula for estimating the number of relocations for import containers when a container is retrieved randomly, and showed that his formula outperforms Watanabe's [1] method in its accuracy. To minimize the number of relocations, Kim et al. [4] suggested a mathematical model and a dynamic programming technique for locating export containers. They also provided decision trees for locating export containers. They assumed that the containers are classified into three groups, according to their weight. Heavier containers tend to be loaded earlier into vessels. Thus, when containers arrive at the yard, placing heavier containers onto higher tiers will reduce the expected number of relocations. Avriel et al. [5] also addressed the relocation (shifting) problem to solve the problem of stowing containers into a vessel. They proposed a 0-1 integer-programming model and a heuristic method for solving the mathematical model.

Although there have been previous studies that addressed the relocation problem, no previous study—except the study by Kim et al. [4]—addressed the problem of determining storage locations considering relocations. Although Kim et al. suggested a method for locating export containers, the case in their study was a special one—which had three groups of blocks—of the case addressed in this study. Also, Kim et al. studied the problem of locating arriving blocks, while this study dealt with the problem of relocating blocks.

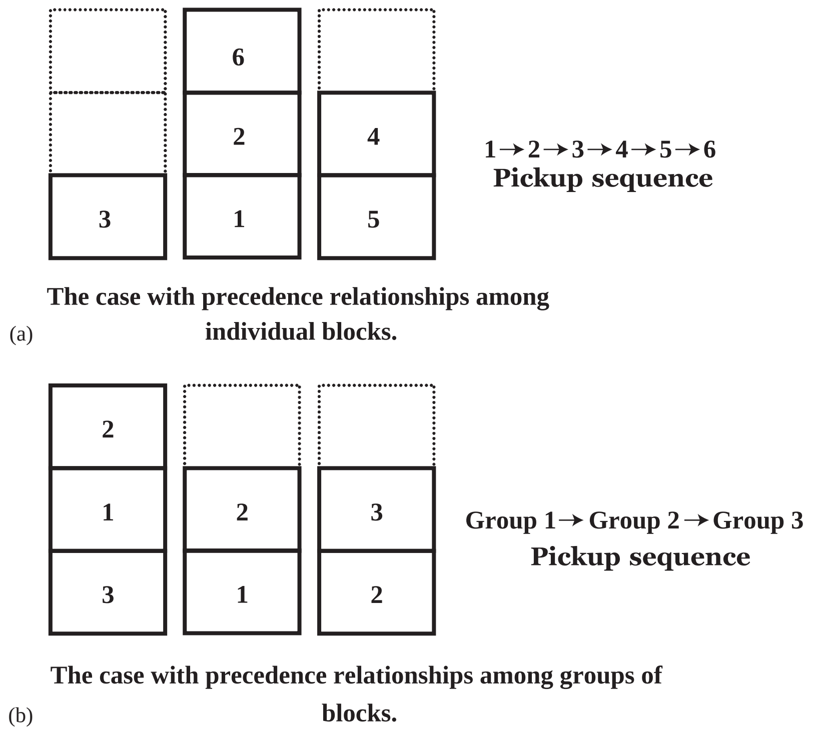
6
2
4
3
1
5
1→2→3→4→5→6
Pickup sequence
The case with precedence relationships among individual blocks.
(a)
2
1
2
3
3
1
2
Group 1→Group 2→Group 3
Pickup sequence
The case with precedence relationships among groups of blocks.
(b)

Fig. 2. Two types of precedence structures: (a) the case with precedence relationships among individual blocks, (b) the case with precedence relationships among groups of blocks.

This study addresses the case in which the pickup precedence among individual blocks is serial, as shown in Fig. 2(a), in which the precedence structure among individual blocks is defined as a chain. Also, this study addressed the case where the precedence structure among groups of blocks can be defined as a chain. The latter case is illustrated in Fig. 2(b), in which blocks with a number of 1 (blocks in group 1) have the highest priority among all the blocks and blocks in group 2—which are numbered 2—have the next priority, and so on.

A practical example of the latter case is as follows: when export containers are to be loaded onto a vessel, because the stowage plan specifies only attributes (destination port, size, and weight class) of containers to be loaded onto a cluster of slots, any container with specified attributes can be loaded into any slot in the cluster. However, due to the structure of a vessel, a precedence relationship exists among clusters of empty slots in the vessel. All clusters of slots in hold must be filled with containers before the loading operation begins for clusters on deck. Also, clusters in lower tiers must be filled with containers before the loading operation for clusters in higher tiers begins.

For a given configuration of a bay, there are three different types of relocations. According to the configuration of the bay in Fig. 2(a), two relocations are already confirmed from blocks 2 and 6 on block 1. Relocations confirmed like this were termed “confirmed relocations.” When block 1 is picked up, blocks 2 and 6 are relocated. Then, “confirmed relocations” for blocks 2 and 6 become “realized relocations.” Although there is no confirmed relocation in stack 1, there is a possibility of having relocations in stack 1,

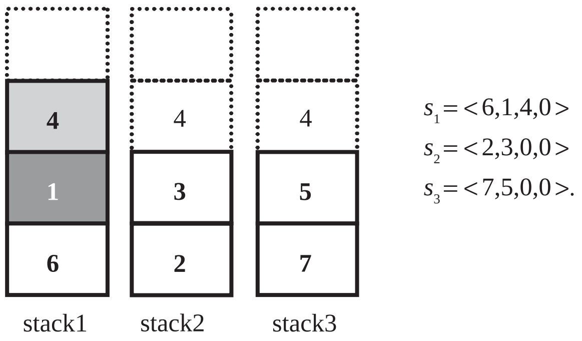

Fig. 3. An illustration of stacks and their states.

considering that future expected blocks may be relocated from other stacks to the second and third tiers of stack 1. This is called "expected additional relocations."

This paper proposes methods for determining the storage locations of relocated blocks and for determining the blocks to be retrieved among multiple blocks with the same priority. This study provides a B&B algorithm for obtaining the optimal locations for relocated blocks and then suggests a heuristic rule for the decision. Also, the performance of the heuristic rule is compared with that of the B&B algorithm.

Section 2 introduces the B&B algorithm for determining the optimal locations of relocated blocks. Section 3 provides a formula for estimating an additional number of relocations and suggests a heuristic rule by using the estimator of the additional number of relocations. Section 4 compares the performance of the heuristic rule with that of the B&B algorithm. Section 5 presents some concluding remarks.

2. A branch and bound algorithm

The following notations will be used to describe the algorithm:

N	: The number of blocks in the initial bay.
r	: The number of stacks in the bay.
s_i^k	: The state (configuration) of stack i after k blocks are picked up from the initial bay. Fig. 3 illustrates a bay with three stacks and the state of each stack.
S^k	: The state of the bay after k blocks are picked up from the initial bay.
a^k	: The action taken when the kth block is picked up. The action accompanies decisions on the block to be picked up (the target block) and the storage locations of relocated blocks. Note that block 1 in Fig. 3 is the target block.
$h(a^k \mid S^{k-1})$	: The number of relocations experienced during action a^k on the bay of state S^{k-1}.
$F(S^k)$	: The minimum total number of relocations to pick up the remaining $N-k$ blocks from the bay of state S^k.

Then, the problem can be defined to be

$$F(S^0) = \underset{a^1}{Min}\{h(a^1 \mid S^0) + F(S^1)\},$$

where S^0 changes to S^1 by action a^1 ($S^0 \overset{a^1}{\rightarrow} S^1$).

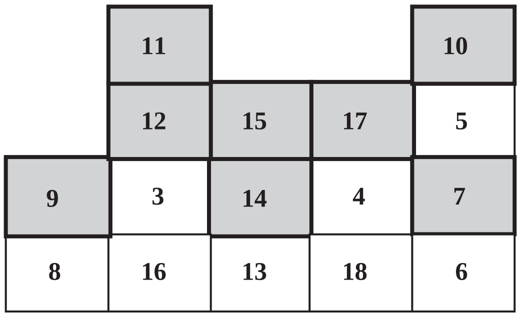

Fig. 4. An illustration of confirmed relocations.

Thus,

$$F(S^0) = \underset{a^1,a^2,\ldots,a^k}{Min} \left\{ \sum_{i=1}^{k} h(a^i|S^{i-1}) + F(S^k) \right\} \quad \text{where } S^{i-1} \overset{a^i}{\rightarrow} S^i \quad \text{for } i = 1, 2, \ldots, k.$$

This problem is solved by using a branch and bound (B&B) algorithm. The initial node in level 0 (the root node) corresponds to the initial state (S^0) of the bay, while nodes in level i correspond to all the possible S^i.

This section proposes B&B algorithms for both cases with precedence relationships among individual blocks and cases with precedence relationships among groups of blocks.

2.1. Cases with precedence relationships among individual blocks

Fig. 2(a) illustrates a case with precedence relationships among individual blocks. The configuration of the bay in Fig. 2(a) can be denoted as [⟨3, 0, 0⟩, ⟨1, 2, 6⟩, ⟨5, 4, 0⟩]. Then, the branches that follow directly the bay in Fig. 2(a) are [⟨3, 6, 2⟩, ⟨0, 0, 0⟩, ⟨5, 4, 0⟩], [⟨3, 6, 0⟩, ⟨0, 0, 0⟩, ⟨5, 4, 2⟩], and [⟨3, 2, 0⟩, ⟨0, 0, 0⟩, ⟨5, 4, 6⟩].

To select the next node for branching, the depth-first and backtracking strategy is used. That is, among unexplored nodes, the node in the highest level is selected for the next branching node. If there is more than one unexplored node in the highest level, then the unexplored node with the minimum lower bound is selected as the next branching node. The depth-first and backtracking strategy was used because excessive computer memory was required when the node with the minimum lower bound was selected as the next branching node from all the levels—which is the usual way of selecting the next node in B&B algorithms.

The lower bound of a node is calculated by adding the cumulative number of realized relocations, from the root node to the current node, to the number of confirmed relocations of the current node. The number of confirmed relocations is obtained by counting blocks located above a block with a higher priority. Fig. 4 illustrates the confirmed relocations, which are illustrated by the shaded blocks.

The B&B algorithm may be improved by defining some dominance rules between the configurations generated from the same configuration in the previous level. For instance, consider a configuration [⟨1, 3, 0, 0⟩, ⟨2, 0, 0, 0⟩, ⟨8, 7, 0, 0⟩, ⟨6, 5, 4, 0⟩]. Then, the following two configurations are examples of possible configurations in the next level, which can be generated from the above configuration: [⟨0, 0, 0, 0⟩, ⟨2, 3, 0, 0⟩, ⟨8, 7, 0, 0⟩, ⟨6, 5, 4, 0⟩] and [⟨0, 0, 0, 0⟩, ⟨2, 0, 0, 0⟩, ⟨8, 7, 3, 0⟩, ⟨6, 5, 4, 0⟩]. It is obvious that the latter dominates the former in the number of relocations. However, the number of nodes that can be pruned by one of these dominance rules is small. Considering the extra computational effort required for applying these dominance rules, the dominance rules were not included in the suggested B&B algorithm.

2.2. Cases with precedence relationships among groups of blocks

The procedure of the B&B algorithm for this case is the same as that for the case in Section 2.1. However, when a node being branched, all the possible configurations resulting from picking up every block with the next highest priority must be enumerated in the next level.

3. A heuristic rule for relocating blocks

When a decision for determining the locations of relocated blocks must be made in real time, time-consuming procedures like the B&B method may not be appropriate in practice. This section proposes a heuristic rule that can be used in real time.

For suggesting a heuristic rule, the concept of "expected number of additional relocations" (ENAR) is useful. ENAR of a stack is the expected number of relocations to be added, considering expected future relocations of blocks from the other stacks in the same bay to the empty spaces of the corresponding stack. The following sub-section presents a method for estimating ENAR.

3.1. Estimating ENAR for cases with precedence relationships among individual blocks

To derive a procedure to estimate ENAR, two assumptions are introduced: one is "optimistic", and the other is "pessimistic." The optimistic assumption is that blocks relocated from a stack will not be relocated again. This assumption is optimistic because it is in fact possible for the relocated blocks from a stack to be relocated again. The pessimistic assumption is that random blocks from other stacks will be relocated to a stack under consideration. This assumption is pessimistic because locations of relocated blocks will be determined in a way of minimizing the number of relocations, the expected number of relocations will be lower than that by the random selection.

The following notations were used to derive expressions of ENAR:

T	: The expected maximum height of a stack in the future.
n	: The highest priority number in a stack.
k	: The number of empty slots in a stack under the assumption that blocks will be filled up to tier T.
$p(i)$	: The probability that blocks with priority i will be stacked into the empty slots of a stack.
$E(k,n)$	: The expected additional relocations resulting from future blocks relocated to the empty slots of a stack with blocks whose highest priority is n and empty slots which number k.

Consider the stack in Fig. 5(a). Suppose that the total number of blocks in a bay (N) is 20. Then, the following shows how to evaluate the value of ENAR resulting from the future relocations into a single empty slot on the top of the stack in Fig. 5(a). If the priority of the block relocated to the empty slot is

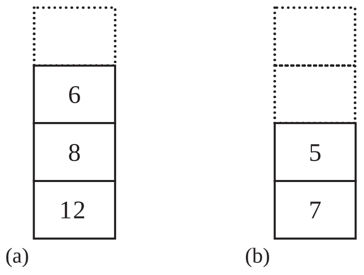

Fig. 5. (a and b) Illustrations of stacks with precedence relationships among individual blocks.

higher than 6, then there will be no relocation. However, if the relocated block has a priority lower than 6, then, a relocation occurs.

Thus, $E(1,6)=$ (probability that the relocated block has a priority higher than 6) $\times$ (the number of relocations (0)) + (the probability that the relocated block has a priority lower than 6) $\times$ (the number of relocations (1))

$$=\frac{5}{17}\times 0+\frac{12}{17}\times 1=\frac{12}{17}.$$

Generally expressed, $E(1,n)$

$$\begin{aligned}&=\frac{n-1}{N-(T-1)}\times 0+\frac{N-(T-1)-(n-1)}{N-(T-1)}\times 1\\&=\frac{n-1}{N-T+1}\times 0+\frac{N-T-n+2}{N-T+1}\times 1\\&=\frac{N-T-n+2}{N-T+1}.\end{aligned}$$

Next, consider the stack in Fig. 5(b). Let $N=10$ and $T=4$. Then,

$$\begin{aligned}E(2,5)={}&p(1)\times\{0+E(1,1)\}+p(2)\times\{0+E(1,2)\}+p(3)\times\{0+E(1,3)\}\\&+p(4)\times\{0+E(1,4)\}+\{p(6)+p(8)+p(9)+p(10)\}\times\{1+E(1,5)\}.\end{aligned}$$

If the block of priority 1 is moved to this stack, no relocation is added by the block, and the resulting stack comes to have $n=1$ and $k=1$. Thus, the first term of the above equation, $p(1)\times\{0+E(1,1)\}$, is the result. However, when the priority of a relocated block is lower than 5, then a relocation must be added, with the resulting stack of $n=1$ and $k=5$. Thus, the final term of the above equation is the result.

Generalizing the above equation results in the following:

$$E(2,n)=\frac{\sum_{i=1}^{n-1}E(1,i)}{N-(T-2)}+\frac{N-n-(T-2-1)}{N-(T-2)}\times\{1+E(1,n)\}.$$

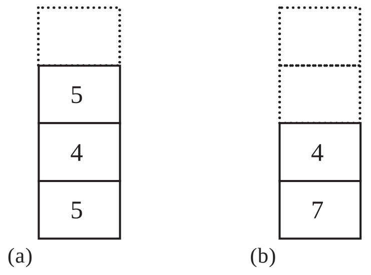

Fig. 6. (a and b) Illustrations of stacks with precedence relationship among groups of blocks.

The above equation can be generalized further to obtain the following equation:

$$\begin{aligned} E(k,n) &= \frac{\sum_{i=1}^{n-1} E(k-1,i)}{N-(T-k)} + \frac{N-n-(T-k-1)}{N-(T-k)} \times \{1+E(k-1,n)\} \\ &= \frac{\sum_{i=1}^{n-1} E(k-1,i)}{N-T+k} + \frac{N-n-T+k+1}{N-T+k} \times \{1+E(k-1,n)\}. \end{aligned} \quad (1)$$

Note that $E(k,n)$ can be calculated recursively, starting from $k=1$.

T is the expected maximum height of a stack in the future and indicates how many empty slots should be considered for estimating the expected number of future relocations. If the average height of stacks in a bay is high, then more empty spaces in the stack under consideration may be filled in the future. However, in the opposite case, fewer empty slots will be used. Thus, in this study, the average height of stacks is used as the value of T. However, it is also necessary to consider stacks higher than the average height of stacks as a candidate for the storage of the relocated blocks. Thus, in order to estimate the value of ENAR for stacks higher than the average height, it is assumed that one empty slot is available even in such stacks. However, if the height of a stack is equal to the maximum allowed height, then it is excluded from consideration, which means that calculating ENAR is not necessary.

3.2. *Estimating ENAR for cases with precedence relationships among groups of blocks*

Consider a stack in Fig. 6(a). Let $N=13$ and the lowest priority of a group be 7. Then,

$$\begin{aligned} E(1,4) &= \{p(1)+p(2)+p(3)+p(4)\} \times 0 + \{p(5)+p(6)+p(7)\} \times 1 \\ &= [13 - \{\text{no. of blocks with priority of 1}\} - \{\text{no. of blocks with priority of 2}\} \\ &\quad - \{\text{no. of blocks with priority of 3}\} - \{\text{no. of blocks with priority of 4}\} - 3] \\ &\quad \div (13-3) \times 1. \end{aligned}$$

This can be generalized to

$$E(1,n) = \frac{N-(T-1)-\sum_{i=1}^{n} n_i}{N-(T-1)},$$

where n_i is the number of blocks with priority i among N blocks in the bay excluding blocks with priority i in the current stack.

Suppose that $N = 13$, the lowest priority group is 7, and the configuration of a stack is as shown in Fig. 6(b). Then,

$$\begin{aligned}E(2,4) = p(1)\times\{0+E(1,1)\} + p(2)\times\{0+E(1,2)\} + p(3)\times\{0+E(1,3)\}\\ + p(4)\times\{0+E(1,4)\} + \{p(5)+p(6)+p(7)\}\times\{1+E(1,4)\}.\end{aligned}$$

This can be generalized as follows:

$$E(2,n) = \frac{\sum_{i=1}^{n}\{n_i \times E(1,i)\}}{N-(T-2)} + \frac{N-(T-2)-\sum_{i=1}^{n} n_i}{N-(T-2)} \times \{1+E(1,n)\}.$$

This can be further generalized as follows:

$$\begin{aligned}E(k,n) &= \frac{\sum_{i=1}^{n}\{n_i \times E(k-1,i)\}}{N-(T-k)} + \frac{N-(T-k)-\sum_{i=1}^{n} n_i}{N-(T-k)} \times \{1+E(k-1,n)\}\\ &= \frac{\sum_{i=1}^{n}\{n_i \times E(k-1,i)\}}{N-T+k} + \frac{N-T+k-\sum_{i=1}^{n} n_i}{N-T+k} \times \{1+E(k-1,n)\}.\end{aligned} \tag{2}$$

3.3. A heuristic rule for determining the storage locations of relocated blocks

The following notations were used for describing the heuristic rule:

S	: The state before relocation. $S=[s_1, s_2, \ldots, s_r]$.
S'	: The state after relocation. $S'=[s'_1, s'_2, \ldots, s'_r]$.
$E(s_i)$	: The total expected additional relocations from stack i in state s_i. This can be calculated by (2).
$E(S)$	: The total expected additional relocations in a bay of state S. $E(S)=\sum_i E(s_i)$.
$r[a]$	: The number of relocations realized or confirmed by action a. An action in this study is defined as relocating blocks on a target block and picking up a target block.
$R[a]$	: The contribution of action a to the total number of relocations. This can be evaluated by $E(S')-E(S)+r[a]$, where the state of a yard-bay, S, is changed into S' by taking action a.

3.3.1. Cases with precedence relationships among individual blocks

When there is a precedence relationship among individual blocks, the next target block is already known. Thus, an action is specified by the locations of relocated blocks. Because this paper assumes that decisions regarding relocations are sequentially made, from the block on the highest tier to the block just on the target block, an action is specified by the location of the next relocation. In this case, an action can be represented by $a=(i, j)$, where i is the next block number to be relocated and j is the location (stack number) to which block i is be relocated.

The rule suggested in this study is to choose the action with the minimum $R[(i, j)]$. Suppose that block i is located in stack k. Because there are changes in the values of $E(s)$ only in stacks k and j,

$$R[(i,j)] = E(s'_k) - E(s_k) + E(s'_j) - E(s_j) + r[(i,j)].$$

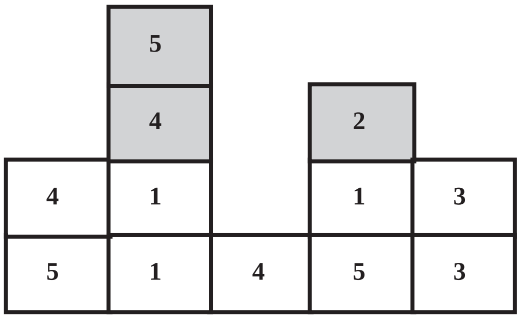

Fig. 7. An illustration of a bay with precedence relationships among groups of blocks.

Because $E(s'_k) - E(s_k)$ is the same for all actions, j, the following results:

$$\underset{j}{Min}\ R[(i,j)] = \underset{j}{Min}\{E(s'_j) - E(s_j) + r[(i,j)]\}.$$

The following illustrates the procedure for determining the location of a relocated block. Consider the stacks in Fig. 3 whose states are

$$s_1 = \langle 6,1,4,0\rangle,\ s_2 = \langle 2,3,0,0\rangle, \quad \text{and} \quad \langle 7,5,0,0\rangle.$$

Suppose that block 1 is to be retrieved from stack 1. Then, block 4 must be relocated to either of the other stacks, 2 or 3. If block 4 is relocated to stack 2, then the state of stack 2 becomes $s'_2 = \langle 2,3,4,0\rangle$. If block 4 is relocated to stack 3, then the state of stack 3 becomes $s'_3 = \langle 7,5,4,0\rangle$. Thus,

$$\begin{aligned} R[(4,2)] &= \{E(s'_2) - E(s_2) + r[(4,2)]\} = \{E(\langle 2,3,4,0\rangle) - E(\langle 2,3,0,0\rangle) + r[(4,2)]\} \\ &= E(1,2) - E(2,2) + 2 \end{aligned}$$

and

$$\begin{aligned} R[(4,3)] &= \{E(s'_3) - E(s_3) + r[(4,3)]\} = \{E(\langle 7,5,4,0\rangle) - E(\langle 7,5,0,0\rangle) + r[(4,3)]\} \\ &= E(1,4) - E(2,5) + 1. \end{aligned}$$

Then, stack j with a lower value of $R[(4,j)]$—between stacks 2 and 3—will be selected as the storage location of relocation for block 4.

3.3.2. Cases with precedence relationships among groups of blocks

Suppose that blocks numbered 1 must be retrieved from the bay in Fig. 7. Although there are two blocks with priority 1 in stack 2, there is no reason to pick up the target block in a lower tier earlier than the target block in a higher tier. Thus, there are two target blocks of priority 1 to be considered as candidates, and one of the two must be selected for the next pickup.

An action in this case is specified by (block ID, location to be relocated) for all blocks stacked on a target block. For example, {(5, 3), (4, 1)} is an action. Although there are only four possible actions regarding the target block in stack 4, the target block in stack 2 has 16 possible actions. To select the next action, this study selects one candidate action among all possible actions for each target block. Then, among them, the action with the least $R[a]$ is selected as the next action. However, when there are more than one block to be relocated for picking up a target block, there may be too many possible actions to be evaluated. Thus, in such a case, this study reduces the number of possible actions for a target block by sequentially finding storage locations for relocating blocks. For example, for target block 1 of stack 2 in Fig. 7, the storage location for relocating block 5 is determined first in the same way as that for the case with precedence relationships among individual blocks, and then, using the resulting configuration of the bay, the storage location for relocating block 4 is determined in the same way. In this way, each target block comes to have only one candidate action.

Suppose that action $\{(5,3),(4,1)\}$ was selected as the candidate action for a target block in stack 2. Then, $S=[\langle 5,4,0,0\rangle, \langle 1,1,4,5\rangle, \langle 4,0,0,0\rangle, \langle 5,1,2,0\rangle, \langle 3,3,0,0\rangle]$ and $S'=[\langle 5,4,4,0\rangle, \langle 1,0,0,0\rangle, \langle 4,5,0,0\rangle, \langle 5,1,2,0\rangle, \langle 3,3,0,0\rangle]$. Then, it is easy to evaluate $R[\{(5,3),(4,1)\}]=E(S')-E(S)+r(a)$. Note that $r(a)$ is the number of realized relocations (2 from realized relocations in the second stack) plus the number of confirmed relocations (1 from a confirmed relocation resulting from moving block 5 to the third stack). Thus, $r(a)$ becomes 3. In the same way, $R[a]$ of the candidate action for retrieving block 1 in stack 4 is calculated. The two $R[a]$s are compared with each other. Block 1 with a lower $R[a]$ will be selected as the next target block. Once the target block is selected, blocks on the top of the selected target block must be relocated in the same way as determined in the process for evaluating $R[a]$.

4. Experiment to evaluate the performance of the heuristic rule

This section compares the performance of the heuristic rule with that of the B&B algorithm. The comparison was performed for both the case with precedence relationships among individual blocks and the case with precedence relationships among groups of blocks. Also, the effects of the bias among the proportions of groups on the performance of the heuristic rule were discussed. The program for the experiment was constructed by Visual C++ 6.0 and run on a personal computer of Pentium III-800 and RAM-128 Mb.

4.1. An experiment for the case with precedence relationships among individual blocks

Forty problems were randomly generated for each combination of number of tiers and stacks in a yard-bay. The average value of the 40 results is provided in each entry in the second and the third columns of Table 1. Table 1 shows that the average error rate of the heuristic rule—calculated as the average of (the value found by the heuristic rule—the value found by the B&B)/(the value found by the B&B)—is 0.073. Table 1 also shows that the number of relocations increases as the number of tiers multiplied by the number of stacks increases. However, because the number of relocations is more sensitive to the number of tiers than to the number of stacks, the trend is not monotonic to the product of the numbers of tiers and stacks.

Table 1 also compares the computational time between the B&B algorithm and the heuristic rule. The computational time for the heuristic rule does not change as the number of tiers or the number of stacks increases, while the computational time for the B&B algorithm increases dramatically when the product of the numbers of tiers and the number of stacks exceeds about 25. Note that the number of stacks in a bay and the number of tiers of a stack of marine containers in port container terminals have been increasing up to the level of 10 and 7, respectively. Also, the number of tiers of a stack of steel plates is much higher than that of marine containers. Also, note that the decision on the locations of relocated blocks must be

Table 1
The comparison between the B&B and the heuristic rule for the case with precedence relationships among individual blocks

No. of tiers × no. of stacks	The average total number of relocations				Average computation time (s)	
	B&B (a)	Heuristic rule (b)	(b)/(a) (%)	Standard deviation of (b)/(a)	B&B	Heuristic rule
3 × 3	3.45	3.55	102.9	0.209	7.3	1.0
3 × 4	5.10	5.25	102.9	0.094	8.0	1.0
3 × 5	7.23	7.43	102.8	0.046	13.2	1.0
3 × 6	8.35	8.65	103.6	0.068	14.5	1.0
3 × 7	9.88	10.35	104.8	0.047	45.5	1.0
3 × 8	11.33	11.80	104.1	0.062	142.3	1.0
4 × 4	9.53	10.33	108.4	0.121	14.8	1.0
4 × 5	11.85	12.63	106.6	0.115	33.4	1.0
4 × 6	13.75	14.78	107.5	0.092	152.4	2.0
4 × 7	16.70	18.28	109.5	0.085	226.8	2.0
5 × 4	12.63	14.13	111.88	0.1	49.5	1.0
5 × 5	15.78	18.03	114.26	0.104	223.3	2.0
5 × 6	21.00	24.38	116.09	0.09	2657.5	2.0

made in real time as soon as a block is requested to be picked up. Thus, these results of the computational time imply that the heuristic rule can be used in real time, while the B&B algorithm is not appropriate for real time usage. When the product of the numbers of tiers and the number of stacks exceed 30, the B&B method could not calculate the number of relocations because of the excessive computational time.

4.2. An experiment for the case with precedence relationships among groups of blocks

For the experiment, three groups were assumed and the number of blocks in a group was generated randomly. Forty problems were generated for each combination of the number of tiers and the number of stacks, except the case with three tiers and seven stacks, for which 20 problems were solved, because of the excessive computational time of the B&B algorithm.

Table 2 shows the average total number of relocations calculated by the B&B algorithm and the heuristic rule. The results show that the average error rate is 4.7%. Note that the average numbers of relocations were significantly reduced, compared with the case of precedence relationships among individual blocks.

Table 2 also shows that the B&B algorithm required much more computational time than did the heuristic rule. Also, the computational time of the B&B algorithm dramatically increased when the product of the number of tiers and stacks approached 25. Note that, when the B&B was used, the computational times in Table 2 were greater than those in Table 1, which implies that the complexity of the problems with precedence relationships among groups is greater than that of problems with precedence relationships among individual blocks. However, on average, the heuristic rule required less than 2 s which is the same level as that for the case with precedence relationships among individual blocks.

4.3. The bias in the proportions of groups

In the experiments for Table 2, the proportions of three groups were assumed to be the same. An experiment was performed to evaluate the effects of the bias in the proportion of groups on the performance

Table 2
The comparison between the B&B and the heuristic rule for the case with precedence relationships among groups

No. of tiers × no. of stacks	The average total number of relocations				Average computation time (s)	
	B&B (a)	Heuristic rule (b)	(b)/(a) (%)	Standard deviation of (b)/(a)	B&B	Heuristic rule
3 × 3	1.58	1.65	104.43	0.101	12.2	1.0
3 × 4	2.70	2.73	101.11	0.032	48.3	1.0
3 × 5	3.78	3.85	101.85	0.061	62.2	1.0
3 × 6	4.68	4.75	101.50	0.039	129.2	1.0
3 × 7	6.67	6.73	100.90	0.065	1927.9	1.0
4 × 4	4.58	4.83	105.46	0.122	55.9	1.0
4 × 5	7.08	7.45	105.23	0.072	339.4	1.0
5 × 4	7.50	8.15	108.67	0.101	132.5	1.0
5 × 5	9.65	10.90	112.95	0.112	1305.8	2.0

Table 3
Percentages of groups for different values of the shape parameter

i	$t=0.25$		$t=0.50$		$t=0.75$		$t=1.00$	
	Y	Percent. (%)	Y	Percent. (%)	Y	Percent. (%)	Y	Percent. (%)
1	1.00	75.98	1.00	57.74	1.00	43.87	1.00	33.33
2	1.19	14.38	1.41	23.91	1.68	29.91	2.00	33.33
3	1.32	9.64	1.73	18.35	2.28	26.22	3.00	33.33

Table 4
The average percentage of errors in the total number of relocations for various shape parameters

t	Average error
0.25	0.18
0.50	0.42
0.75	1.77
1.00	2.78

of the heuristic rule. Let the cumulative proportion of blocks that are included in group i be $Y = i^t$, where Y, i, and t represent the cumulative proportion of blocks, the index for a group, and the shape parameter of the cumulative proportion curve ($0 \leqslant t \leqslant 1$), respectively.

Table 3 shows the percentage of each group when there are three groups for different values of the shape parameter, t. Note that as the value of the shape parameter becomes smaller, the percentages of blocks become concentrated into one or two groups.

Blocks in randomly generated bays with six stacks and four tiers were retrieved according to a decision by both the B&B algorithm and the heuristic rule. Two hundred eighty different configurations of bays were generated for each value of the shape parameter, 0.25, 0.50, 0.75, and 1.00. The average percentage of errors in the total number of relocations is listed in Table 4. It was observed that the average error increased as the value of the shape parameter increased. It was also observed that when the value of the shape parameter was large, the total number of relocations itself was large, which seemed to partially contribute to the higher average error for the higher value of the shape parameter.

5. Conclusions

This study addressed the problem of relocating blocks in block-stacking warehouses. A B&B algorithm and a heuristic rule based on probability theory were proposed. A procedure for estimating the expected additional number of relocations was suggested for various configurations of stacks.

A numerical experiment compared the performance of the heuristic rule with that of the B&B algorithm. The comparisons showed that the total number of relocations calculated by the heuristic rule exceeded that found by the B&B algorithm by an average of approximately 7.3% for the case with precedence relationships among individual blocks and 4.7% for the case with precedence relationships among groups of blocks. Also, the difference increased as the numbers of tiers and stacks increased. For the same sized bay, the average number of relocations for the case with precedence relationships among individual blocks was larger than those for the case with precedence relationships among groups. The computational time of the heuristic rule was less than 2 s—which is within the level that can be used in real time. In contrast, when the product of the numbers of tiers and stacks exceeded 20, the computational time of the B&B algorithm exceeded 100 s for the case with precedence relationships among individual blocks and 1000 s for the case with precedence relationships among groups—which exceed the levels that can be used in real time. Also, for the case with precedence relationships among groups, it was found that, as the bias in the percentages of blocks among blocks increased, the difference in the total number of relocations between the B&B algorithm and the heuristic rule decreased.

This study addressed blocks stacked on a floor. However, there are other types of products that require relocations and that are stacked in various different forms. Examples are steel coils, paper rolls, steel plates, boxes on drive-in racks, and so on. Each of them requires a different analysis that considers unique physical characteristics of stacking. Decision-making problems for locating and relocating these other types of block stacking products will be challenging topics for future studies.

Acknowledgements

This study is a part of the results of overseas research supported by Pusan National University in 2002.

References

[1] Watanabe I. Characteristics and analysis method of efficiencies of container terminal—an approach to the optimal loading/unloading method. Container Age, March 1991, p. 36–47.

[2] Castilho B, Daganzo CF. Handling strategies for import containers at marine terminals. Transportation Research 1993;27B(2):151–66.

[3] Kim KH. Evaluation of the number of rehandles in container yards. Computers and Industrial Engineering 1997;32(4): 701–11.

[4] Kim KH, Park YM, Ryu KR. Deriving decision rules to locate export containers in container yards. European Journal of Operational Research 2000;124:89–101.

[5] Avriel M, Penn M, Shpirer N, Witteboon S. Stowage planning for container ships to reduce the number of shifts. Annals of Operations Research 1998;76:55–71.