Int. J. Production Economics 135 (2012) 73–80

Contents lists available at ScienceDirect

Int. J. Production Economics

journal homepage: www.elsevier.com/locate/ijpe

The storage location assignment problem for outbound containers in a maritime terminal

Lu Chen*, Zhiqiang Lu

School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

ARTICLE INFO

Article history:
Received 22 October 2009
Accepted 16 September 2010
Available online 25 September 2010

Keywords:
Maritime terminal
Container
Storage location assignment
Mixed integer programming
Heuristics

ABSTRACT

This paper addresses the storage location assignment problem for outbound containers. The problem is decomposed into two stages. The yard bays and the amount of locations in each yard bay, which will be assigned to the containers bounded for different ships, are determined in the first stage. The exact storage location for each container is determined in the second stage. The problem in the first stage is solved by a mixed integer programming model, while a hybrid sequence stacking algorithm is applied to solve the problem in the second stage. Experimental results show that the proposed approach is effective and efficient in solving the storage location assignment problem for outbound containers.

1. Introduction

A container terminal is an inter-modal interface in the global transportation network. Containers are stored temporarily to account for the differences in arrival times of the sea and land carriers. Storage location assignment for arriving containers is important in improving the efficiency of container handling and reducing the turnaround time of a ship.

Inbound and outbound container operations are different. Inbound containers arrive predictably in large batches at yard, but depart one by one in an unpredictable order when they are claimed. Outbound containers depart predictably but arrive in a random order. They must be loaded according to a rigid ship storage plan, in order to maintain the stability of the ship, and satisfy the loading requirement that is specified by destination and size of containers.

This paper focuses on the operational decision making problem in stacking outbound containers. The remaining part of this paper is organized as follows. In Section 2, we give a brief review of previous work in the area of container storage location assignment. Then a detailed problem description is given and a solution approach is outlined in Section 3. With this approach, the problem is modeled and solved in two stages: Sections 4 and 5 for the first and the second stages, respectively. Computational experiments are conducted for realistic settings and the results are reported in Section 6. Finally, in Section 7 we present our conclusions and perspectives.

* Corresponding author. Tel.: +86 21 3420 7010; fax: +86 21 3420 6782.
E-mail address: chenlu@sjtu.edu.cn (L. Chen).

0925-5273/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.ijpe.2010.09.019

2. Literature review

Dekker et al. (2006) explored different stacking policies for containers in automated terminals by means of simulation. A comprehensive overview of stacking policies used in practice is provided. Zhang et al. (2003) studied the storage space allocation problem in the storage yards of terminals. Both inbound containers and outbound containers are considered, and are allowed to be mixed up in one block. They decomposed the space allocation problem into two levels. In the first level, the total number of containers to be placed in each storage block is determined. In the second level, the number of containers associated with each ship is determined to minimize the total transportation distance for moving containers between blocks and vessel berthing locations.

Usually the location assignment strategies are treated differently for inbound and outbound containers. For *inbound containers*, Castilho and Daganzo (1992) presented two strategies in storage space assignment: non-segregating strategy and segregating strategy. In non-segregating strategy, new arriving containers are piled on top of the existing ones by placing them in empty slots until they were filled up to the prescribed level. In segregating strategy, a space is emptied and allocated whenever a batch of newly import containers is unloaded from each ship. But the benefits of segregation must be traded off against the need for clearing moves. Considering the segregating storage strategy for inbound containers, Kim and Kim (1999) thought that one of the difficult problems in yard operation was that it took too much handling effort for yard cranes to rehandle containers on the top of the requested container. The authors derived a formula that describes the relationship between the height of container stacks and the number of rehandles. Kim and Kim (2007) proposed methods for determining prices for the storage of containers in a yard. A storage fee encourages customers to store containers only for a short period of time in the terminal's yard.

For *outbound containers*, Taleb-Ibrahimi et al. (1993) described two different handling and storage strategies: (1) static space allocation strategy, which means that upon arrival, a container is sent to a specific location in storage area and its position is not allowed to vary in the yard during its stay at the terminal. (2) Dynamic strategy, which means containers arrived much earlier than their scheduled departures are stored in a temporary area, until space in the storage area is assigned to them. An operating procedure and a heuristic algorithm were presented to determine the best times for reserving space in the permanent area for various ships. This is different from the approach considered in this paper where storage locations are allocated to each container. Preston and Kozan (2001) used genetic algorithm technique to determine the way that outbound containers are stored to minimize the time spent transferring containers from a storage area to ship or ship to the storage area. It is the sum of setup times (i.e. the time necessary to retrieve containers from the stack) and travel times (i.e. the time necessary to transport containers from the stack to the ship). However, a container is assumed to have a pre-assigned storage area in the terminal, and the random arrival of outbound containers is ignored in this study.

Kim and Park (2003) decomposed the process of determining the storage locations for outbound containers into two stages: space allocation stage and stage of locating individual containers. In the space allocation stage, the amount of space of each block that would be allocated to each ship for future outbound containers is determined. In the stage of locating individual container, a decision on the exact storage location for each outbound container is made whenever an outbound container arrives at the terminal. The authors developed a mixed integer program to solve the space allocation problem in the first stage, with the objectives to minimize the containers' delivery cost between the berth and the yard and to minimize the traveling cost of YCs to pick up the outbound containers within a certain range of yard area. Kim et al. (2000) focused on the stage of locating individual container by determining the storage location of an outbound container in a pre-assigned yard bay in order to reduce rehandle during loading operations. Dynamic programming is used to solve the problem. However, how to select a yard bay for coming outbound containers has not been addressed in both stages. In addition, destination information of containers has not been considered.

It is clear from the literature review that there have been no comprehensive studies on the storage location assignment problem for outbound containers. In this paper, a systematic approach is proposed to determine the storage location for outbound containers in a maritime terminal. The objectives are:

Efficient use of storage space in the yard;
Efficient transportation of outbound containers from yard to berth; and
Minimization of rehandle operation, thus to achieve maximum efficiency in the loading operations.

A rehandle is a container movement made in order to permit access to another container, and is considered as an unproductive move.

3. Problem description and solution approach

3.1. Problem description

We assume that yard cranes and trucks are used as container handling equipment in the yard. When an outside truck delivers an outbound container to the yard, a yard crane picks it up and stacks it in a yard bay. During the loading operation, a yard crane picks up the container and puts it on a yard truck that transfers it to a quay crane.

Every ship which is loaded at a terminal has a storage plan. The shipping line makes a rough plan based on container categories, which is sent to the terminal. Before the arrival of the ship, a more detailed plan is made by the terminal planner who fills the categories with detailed containers. The storage plan specifies which container in the storage yard will be loaded at which location in the ship. The objectives are: (1) to satisfy the ship's stability; and (2) to minimize the handling effort of quay cranes and yard equipment.

Within the storage yard containers are stored in blocks. Each block consists of 20–30 yard bays that are four or five containers high. Each yard bay has six rows side by side (see Fig. 1). Normally, outbound containers start arriving at the terminal waiting for loading 3–4 days before the ship departs. In order to have an efficient loading sequence, outbound containers must be laid out in the optimal locations. However, the ideal layout of outbound containers in the storage yard is almost impossible to be achieved due to the random arrival of containers at the terminal.

A way to improve the efficiency of loading operations would be container shuffling in advance of loading in order to group the containers by destination and weight. However, this task necessitates additional workload for the handling equipment. Therefore, the shuffling of containers could be performed only when the handling equipment is idle. In addition, shuffling may require a buffer stacking area, which seems hardly practical or realistic for land scarce container terminals.

Therefore, determining the storage location for outbound containers becomes crucial in increasing the productivity of the loading operations.

3.2. A two-stage solution approach

The storage location assignment problem concerns the reservation of storage space for outbound containers bounded for each ship and the decision to store a container at a particular location. These two decision making problems are studied respectively in the following two stages:

(1) Yard bay allocation: in the first stage, yard bays and the amount of locations in each yard bay, which will be assigned to outbound containers, are determined. The objectives in this stage are to reduce the time required for the yard trucks to transfer the containers from the yard to the berth for loading onto the ships, and to balance the workload of each yard bays.
(2) Determining the exact location: in the second stage, the exact storage location in the pre-assigned yard bays is determined for each container upon its arrival at the terminal. The objective is to reduce rehandle during the future loading operations.

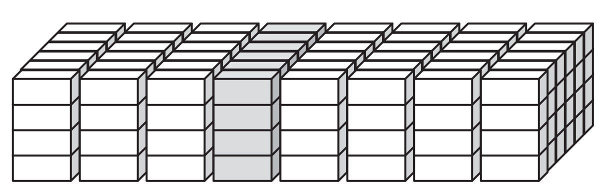

Fig. 1. A container block in storage yard.

4. Yard bay allocation for outbound containers

This section discusses the yard bay allocation problem in the first stage. A mixed integer programming (MIP) model is formulated to solve this problem. The yard bays and the amount of space in each yard bay that will be allocated to store outbound container in each period of the planning horizon are determined.

The following assumptions are made:

(1) Since the scheduling problem of the handling system is beyond the scope of this paper, we assume that there is enough resource to handle the outbound containers in the terminal.
(2) The berth allocation of ships is assumed to be known.
(3) The containers are assumed to be of one size. And the workloads of yard cranes are calculated in terms of the number of containers. Containers with different sizes are normally not mixed in blocks. Therefore, our approach is not directly affected with containers of different sizes, since the storage of containers of different sizes can be separated into independent problems.
(4) The inbound containers and outbound containers are not mixed up in one block in the storage yard.
(5) Usually, there are several calling ports of a ship's voyage itinerary. For outbound containers, containers of different destinations are not mixed up in one yard bay. This principle is to minimize unnecessary rehandle operation, since containers of the same group are likely to be loaded onto slots located close to each other in the ship.

Assumptions (4) and (5) are based on the observation on the reality of terminals in Shanghai.

The following notations are introduced:

K the total number of container blocks for storing outbound containers in the storage yard;
B the total number of yard bays for storing outbound containers;
S_A the set of ships for which space should be allocated during the planning horizon;
S_L the set of ships that will be loaded during the planning horizon;
S_S the set of ships for which space has been allocated during the previous planning horizon; note that ships in S_A may also belong to S_S;
S the set of all the ships during the planning horizon, $S=S_A \cup S_L \cup S_S$
i index of yard bays, $1 \le i \le B$;
j index of ships;
k index of blocks, $1 \le k \le K$;
C_i the storage capacity of yard bay i;
b_i the number of the container block that yard bay i belongs to;
m_j the maximum number of bays that containers for ship j can be stacked into;
D_j the number of destinations of the voyage itinerary of ship j, $j \in S_A$;
V_i^0 the number of containers in yard bay i at the beginning of the planning horizon, $1 \le i \le B$;
N_j the expected number of outbound containers for ship j, which will arrive at the yard during the planning horizon, $j \in S_A$;
B_j the set of yard bays that have been allocated to store containers bounded for ship j, $j \in S \backslash S_A$. Define $\hat{B}=\{B_j | j \in S \backslash S_A\}$, yard bays that cannot be allocated to ships from S_A.
d_{ij} the travel distance of yard vehicles between yard bay i and the berthing location of ship j;

The decision variables are:

x_{ij} the number of containers bounded for ship j, which will be stored in yard bay i during the planning horizon, $j \in S_A$.
$\delta_{ij}=1$ if containers for ship j are stacked in yard bay i; 0, otherwise, $j \in S_A$. For $j \in S \backslash S_A$, δ_{ij} is determined in the previous planning horizons, and is known.
M_j the total number of yard bays assigned to ship j, $j \in S_A$. For $j \in S \backslash S_A$, M_j is determined in the previous planning horizons, and is known.
V_i the total number of containers in yard bay i at the end of the planning horizon.
W_k the workload of block k during the planning horizon.

The space allocation problem can be formulated as follows:

$$\text{Min}\left(w_1 \sum_{i=1}^{B} \sum_{j \in S_A} x_{ij} d_{ij} + w_2 \left(\max_{\{k\}} W_k - \min_{\{k\}} W_k\right)\right) \tag{1}$$

Subject to:

$$V_i = V_i^0 + \sum_{j \in S_A} x_{ij}, \quad i=1,2,\ldots,B \quad \text{and} \quad i \notin \hat{B} \tag{2}$$

$$V_i \le \gamma C_i, \quad i=1,2,\ldots,B \tag{3}$$

$$V_i = 0, \quad i \in B_j, \quad j \in S_L \tag{4}$$

$$\sum_{i=1}^{B} x_{ij} = N_j, \quad j \in S_A \tag{5}$$

$$x_{ij} \le N_j \delta_{ij}, \quad i=1,2,\ldots,B, \quad j \in S_A \tag{6}$$

$$\sum_{j \in S} \delta_{ij} \le 1, \quad i=1,2,\ldots,B \tag{7}$$

$$M_j = \sum_{i=1}^{B} \delta_{ij}, \quad j \in S_A \tag{8}$$

$$M_j \ge D_j, \quad j \in S_A \tag{9}$$

$$M_j \le m_j, \quad j \in S_A \tag{10}$$

$$\sum_{j \in S} M_j \le B \tag{11}$$

$$W_k = \sum_{\substack{j \in S_A \\ b_i = k}} x_{ij}, \quad k=1,2,\ldots,K \tag{12}$$

$$x_{ij}, V_i, M_j, W_k \ge 0, \quad i=1,2,\ldots,B, \quad j \in S_A, \quad k=1,2,\ldots,K \tag{13}$$

$$\delta_{ij} \in \{0,1\}, \quad i=1,2,\ldots,B, \quad j \in S_A \tag{14}$$

The travel distance between the berth and the storage yard for the outbound containers depends on the location of the allocated space for each ship. The travel distance of one container from yard bay i to the berthing location of ship j is d_{ij}. Therefore, the total travel distance is calculated and minimized in the first term in the objective function (1). The second term of (1) measures the imbalance of the number of containers stacked in each yard bay during the planning horizon. w_1 and w_2, the weights of the two terms in (1), are adjusted according to the relative importance of the two objectives. Constraints (2) define the total number of containers in the yard bays at the end of the planning horizon. Constraints (3) ensure that the density of each yard bay will not exceed the allowable level, where γ is the allowable density for each yard bay. The spare space is reserved in practice for possible rehandling. By constraints (4), those bays occupied by the containers bounded for ship j, $j \in S_L$, are emptied, and become available in the next planning horizon. Constraints (5) ensure that the space requirement of each ship during the planning horizon must be satisfied. Constraints (6) define the variable δ_{ij}, such that $\delta_{ij}=1$ if containers of ship j are stacked in yard bay i during the planning horizon. Constraints (7) ensure that the containers bounded for different ships are not mixed up in the same bay. Constraints (8) specify the total number of yard bays that are used for storing the containers for ship j. Constraints (9) ensure that number of yard bays assigned to a ship is greater or equal to D_j, the number of destinations of the voyage itinerary of ship j. This is based on the principle that containers of different destinations are not mixed up in one yard bay. Constraints (10) specify the maximum number of bays that are used for storing the containers for ship j. It can avoid the distribution of containers over too wide areas. Constraints (11) ensure that the total number of yard bays assigned in the planning horizon cannot exceed the total amount of available yard bays in the terminal. Constraints (12) define the workload in each container block, which is evaluated by the total number of containers handled in the planning horizon. Constraints (13) guarantee the non-negative values of the variables. Constraints (14) specify the binary decision variables.

The above model is non-linear because of the objective function. To convert it to a linear model, we define

$$P=\max_{\{k\}} \sum_{\substack{j\in S_A\\ b_i=k}} x_{ij}, \quad Q=\min_{\{k\}} \sum_{\substack{j\in S_A\\ b_i=k}} x_{ij}$$

Then, the objective function can be rewritten as

$$\mathrm{Min}\left(w_1 \sum_{i=1}^{B} \sum_{j\in S_A} x_{ij} d_{ij} + w_2(P-Q)\right)$$

Subject to constraints (2)–(14) and

$$\sum_{\substack{j\in S_A\\ b_i=k}} x_{ij} \le P, \quad k=1,2,\ldots,K \tag{15}$$

$$\sum_{\substack{j\in S_A\\ b_i=k}} x_{ij} \ge Q, \quad k=1,2,\ldots,K \tag{16}$$

The additional constraints (15) and (16) define the new variables P and Q in the model. It is obvious that the new linear model is equivalent to the original one.

5. Determine the storage locations for outbound containers

The first stage determines the space in each yard bay that can be allocated to outbound containers for different ships in each planning period. The remaining problem is to determine the storage location to stack the next arriving container among several empty slots in the pre-assigned yard bays. The objective is to store outbound containers for the final storage layout from which an efficient loading sequence can be constructed.

5.1. Problem description

In this stage, containers are classified into several weight groups, based on container weight. We propose a methodology to determine the storage slot for an arriving container of a weight group in a pre-assigned yard-bay, considering the configuration of containers of different weight groups in the yard bay. The objective is to minimize the expected number of rehandle during loading operations.

It is assumed that within a certain yard bay, heavier containers are to be loaded before lighter containers. In practice, however, there may be some flexibility in loading containers with different weight. For example, when containers in a yard bay are loaded onto a ship, lighter containers are loaded before heavier ones. In general, heavier containers are more likely to be loaded earlier. Therefore, if heavy containers are located in lower tiers in the yard bays, rehandle operations will likely occur. It is reasonable to minimize the rehandle operations under this assumption.

5.2. Sequence stacking

Determining the storage location for an outbound container is a real-time decision making problem. It is difficult to establish a global optimization procedure to solve this problem. Sequence stacking (Hao et al., 2000) is widely used in practice, which means containers of the same weight level are stacked in the same stack or in the same tier.

Fig. 2 shows a bay configuration in sequence stacking. The number in each slot represents the weight level of the container that is stacked in that slot. The bigger the number, the higher is the weight level. Five weight levels and four weight levels are considered, respectively, in vertical stacking and horizontal stacking. The empty slots in both cases are reserved as stacking buffers. Due to the random arrival sequence of containers, it is difficult to obtain the ideal bay configuration shown in Fig. 2. Normally, a lot of rehandle operation is needed in the stacking process. For example, for vertical stacking shown in Fig. 2(a), if a lot of heavy containers arrive firstly at the yard, then the stack near the truck lane would be full. Thus, the movement of the yard crane becomes difficult to stack the upcoming light containers. For horizontal stacking shown in Fig. 2(b), if a heavy container arrives firstly at the yard, it must be put in the buffer slot and re-shuffling is needed when light containers arrive.

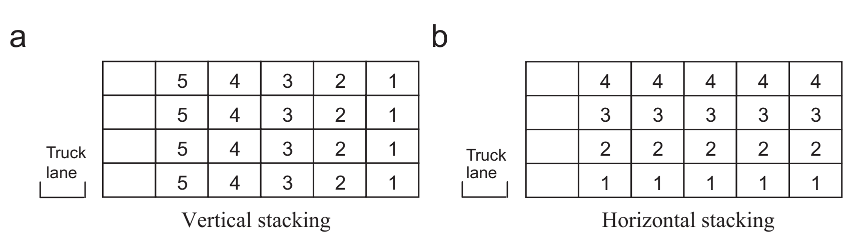

Fig. 2. Sequence stacking.

5.3. Hybrid sequence stacking

In this section, we present a hybrid sequence stacking method to determine the best storage location in a yard bay for an arriving container given an arbitrary configuration of the bay.

Fig. 3 shows an ideal configuration of a bay in hybrid sequence stacking. Similarly, the number in each slot represents the weight level of the container that is stacked in that slot. Here, containers are divided into nine levels according to their weights. The arrows indicate the increase in the direction of containers' weight. Thus, in this configuration of a bay, heavier containers are stacked in left upper locations, and lighter containers are stacked in right lower locations. The containers of the same weight level are not stored in the same stack or in the same tier in this configuration, which allows more optimization space when determining the storage location. Another reason for this principle is that, it is always easier for a yard crane to pick up a container from a slot whose left side is empty.

In Fig. 3, the bay is divided into nine sets, each of which includes the optimal storage locations for the containers with the corresponding weight level. For instance, the set for level 6 has four slots, which are optimal storage locations for containers of weight level 6. Though nine weight levels are considered in this configuration, the method is suitable with little modification for less number of weight levels, i.e. to combine consecutive levels.

5.4. Hybrid sequence stacking algorithm

A hybrid sequence stacking algorithm (HSSA) is described in this section to decide where to put a newly arriving container. The following notations are used in describing the HSSA:

$[W_{min}, W_{max}]$	The weight range of the containers
W_c	The weight of container c
L_c	The weight level of container c
l	Index of a weight level
S_l	The set of optimal slots for containers with weight level l
(x_l, y_l)	The geometric center of the slots in set S_l
(x_c, y_c)	The best storage location for container c
(x_c^s, y_c^s)	The available storage locations, which do not belong to set S_l, for container c

Given the weight of a container c, the weight level L_c is calculated in the following way:

$$W_{min}+(L_c-1)\frac{W_{\max}-W_{max}}{8} \le W_c \le W_{min}+L_c\frac{W_{max}-W_{min}}{8}$$

In order to present the HSSA, we put the bay into a coordinate system as shown in Fig. 4. Each slot represents a point in the coordinate system. From Fig. 4, it is easy to get that $(x_1, y_1)=(6, 1)$, $(x_2, y_2)=(5.5, 1.5)$, $(x_3, y_3)=(5, 2)$, $(x_4, y_4)=(4.5, 2.5)$, $(x_5, y_5)=(3.5, 2.5)$, $(x_6, y_6)=(2.5, 2.5)$, $(x_7, y_7)=(2, 3)$, $(x_8, y_8)=(1.5, 3.5)$, and $(x_9, y_9)=(1, 4)$.

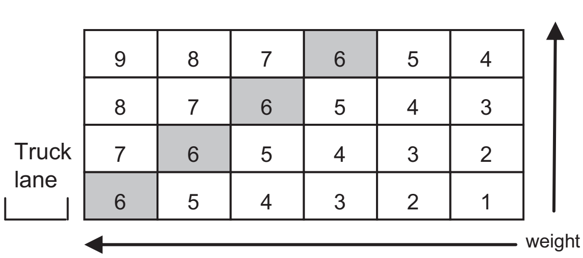

Fig. 3. An ideal configuration of a bay.

The procedure of the HSSA is illustrated in Fig. 5 and the details are elaborated in the following steps:

Step 1: *Select a bay*: select among the yard bays pre-assigned in the first stage a yard bay that is not full and only with containers of the same destination with the arriving container c; if no such bay can be found, the arriving container is stacked in an empty pre-assigned bay.

Step 2: *Get weight level*: get the weight W_c of the arriving container c, calculate the weight level L_c of container c.

Step 3: *Find a slot in the set of optimal storage slots*: get the set of optimal storage slots S_{L_c}, if there exist available slots in S_{L_c} (an empty slot is called "available" when all the slots below the empty slot are occupied), then

(1) Randomly select an available location (x_c, y_c);
(2) Stack container c in (x_c, y_c);

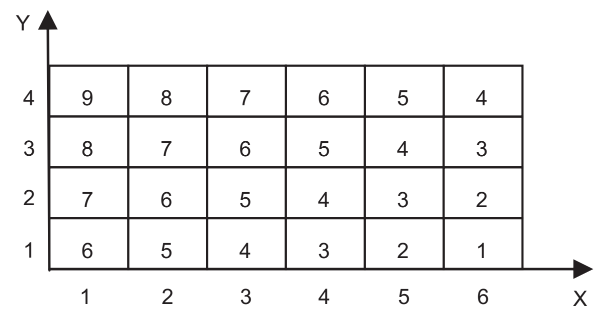

Fig. 4. An illustration of a bay coordinate system.

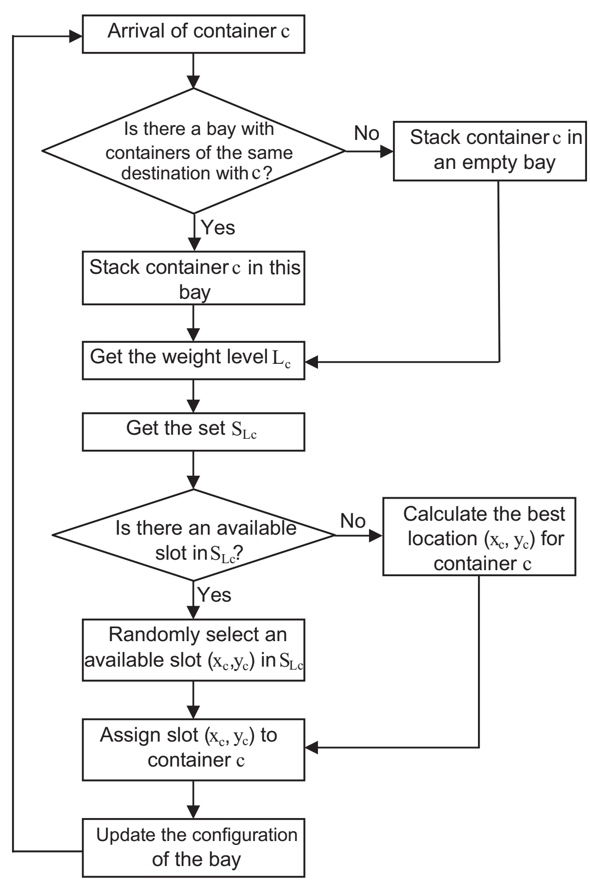

Fig. 5. The HSSA process.

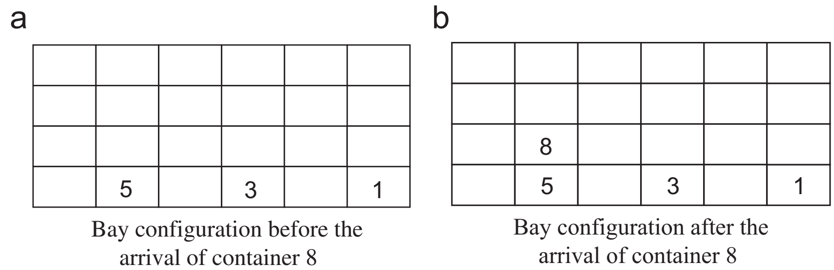

Fig. 6. The bay configuration after the arrival of containers 1, 3, and 5.

(3) Update the configuration of the bay;
(4) Go to Step 1;

Otherwise go to Step 4.

Step 4: *Find a slot in the remaining available slots in the bay*: for each available slot (x_c^s, y_c^s) in the bay, do

(1) Calculate the rectilinear distance from the geometric center of S_{L_c} as

$$d_s = |x_{L_c} - x_c^s| + |y_{L_c} - y_c^s|$$

(2) Select the slot with the minimum distance from the geometric center as the storage location (x_c, y_c);
(3) Stack container c in (x_c, y_c);
(4) Update the configuration of the bay;
(5) Go to Step 1;

In Step 4(2), in case that multiple locations are obtained using the above evaluation function, the optimal location is selected using the following rule. The heavier container is always stacked in the left upper location, and the lighter container is always stacked in the right lower location.

5.5. A numerical example

A numerical example is provided here to illustrate the effectiveness of the HSSA. The yard bay has four tiers and six lanes. We randomly generated an arrival sequence of 18 outbound containers with the same destination port. Let the containers be denoted by their weight level; the arrival sequence (also the stacking sequence) of these containers is: 1, 3, 5, 8, 5, 9, 2, 3, 6, 4, 7, 1, 8, 7, 6, 9, 2, and 4. Containers 1, 3, and 5 are stacked in one of their optimal storage slots. When container 8 arrives, the bay configuration is as shown in Fig. 6(a). It can be observed that the optimal storage slots for container 8 (indicated as shaded slots in the Fig. 6(a)) are not available. Therefore, container 8 is stacked in the slot that is closest to the geometric center of the set of the optimal storage slots for container 8, as shown in Fig. 6(b).

Applying the HSSA described in Section 5.4, for each arriving container, the storage location for each container is shown in Fig. 7. We assume that containers of higher weight level are loaded before containers of lower weight level. The loading sequence is represented by the subscript of each container shown in Fig. 7. Given the loading sequence, only one rehandle operation is needed, that is, container 7_5 needs to be rehandled when handling container 9_2.

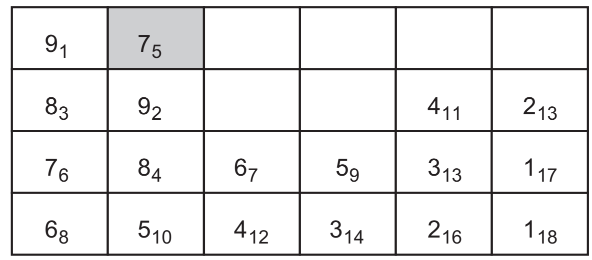

Fig. 7. Storage locations for the containers in the numerical example.

6. Experiments

In this section, the proposed storage location assignment method is evaluated using practical data generated from a typical container terminal in Shanghai. The approach is coded in Visual C++ and run on a personal computer with duo CPU @ 2 GHz and 2 GB RAM. The mixed integer programming model is solved using CPLEX 10.0, a commercial software package.

6.1. Problem settings

In the experiment, there are 10 blocks for stacking outbound containers in the storage yard. Each block has 20 bays. Each bay has 6 stacks that are 4 containers high. As some empty slots in each bay is usually reserved for possible rehandle operation, the actual capacity for use in each bay is $\gamma C_i = 0.8C_i$.

In the first level of the approach, the following container flow data are needed: V_i^0, N_j, D_j, and B_j. In the experiment these data are generated based on the distributions of the real container flows at Shanghai container terminals. For a certain ship, the distribution of numbers of containers arriving during different time period is assumed to be the Poisson distribution.

In the second level of determining the specific location, the following container information is needed: destination and weight of each container. In the experiment, containers are assigned with a destination port randomly chosen from the several calling ports of a ship's voyage itinerary. The distribution of container weight can be estimated from the past empirical data. Container weight information used in the experiment is generated using the distribution. Based on the weight information, containers for each ship are classified into several predetermined weight groups.

6.2. Implementation of the 2-stage approach

The proposed 2-stage approach is tested for a total of 120 planning periods. Let t denote the planning horizon; the procedure for implementing the approach is as follows:

For $t=1$ to 120, do {

1. Generate the input data for the planning horizon t.
2. Solve the models (1)–(16) to obtain the numbers of containers bounded for each ship allocated to each bay.

3. Apply the HSSA to determine the specific locations for containers arriving in period t.
4. Update yard bays information; update V_i^0.

}

6.3. Performance on yard bay allocation

We carried out experiments with different parameter settings, namely the weights in the MIP model. Because of the large size of the model, there is no guarantee that it can be solved to optimality in a short period of time. We retrieve the solution from CPLEX after 30 min for some large sized problems. The computational results of the performance on space allocation for the 120 planning periods are summarized in Table 1.

The two objectives defined in (1) have different value scales. In order that both values can be minimized at the same time, we replace the total travel distance with average travel distance in the objective function of the MIP model. Average travel distance is calculated by dividing the total travel distance with the total number of outbound containers in a planning horizon. We firstly set w_1, the weight of the average travel distance between the yard and the berth, to be 1, and w_2, the weight of the imbalance of number of containers among different blocks, to be 0. We minimize only the average travel distance and ignore the other objective. From the results in Table 1, the model can be solved to optimality for all the testing horizons. And the computation times are very short. However, the performance in terms of the average imbalance in the number of containers is not good, with 65.5 containers in average. Next we set $w_1=w_2=0.5$. In case that the model cannot be solved completely in 30 min, 1800 s of computation time is used to calculate the average value. The average computation time for a planning horizon is 668.5 s, which is short compared to the planning cycle (i.e. 6 h). Although most of the models cannot be solved to optimality with $w_1=w_2=0.5$, the performance in terms of average imbalance in the number of containers is much better. And the performance in terms of average travel distance is slightly worse than the results with $w_1=1$ and $w_2=0$. This shows that combining the two types of objective is a good strategy.

When $w_1=w_2=0.5$, the relative gap to the lower bound has an average of 2.64% and a maximum of 5.33%. In some periods, optimal solution is obtained. Fig. 8 illustrates the relative gap over the 120 periods tested. The imbalance of the number of containers among the 10 blocks is calculated by $\max_{\{k\}} W_k - \min_{\{k\}} W_k$. Fig. 9 illustrates the imbalance of number of containers over the 120 periods tested. The average imbalance of the number of containers among 10 blocks is 22.2. It represents a large improvement over the average figure of 10% of the block capacity (around 40 containers) in the current real operation. The results in Table 1 show the effectiveness of the space allocation approach.

Table 1
Computational results of performance on bay allocation.

Parameter settings	Computation time (s)	Relative gap to lower bound (%)	Imbalance in no. of containers
$w_1=1$, $w_2=0$			
Average	2.5	0	65.5
Maximum	5	0	110
Minimum	0.21	0	25
$w_1=0.5$, $w_2=0.5$			
Average	668.5	2.64	22.2
Maximum	> 1800	5.33	45
Minimum	2.5	0	9

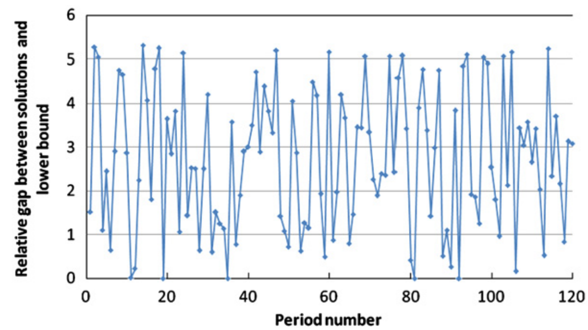

Fig. 8. Relative gap from the lower bound for the total travel distance.

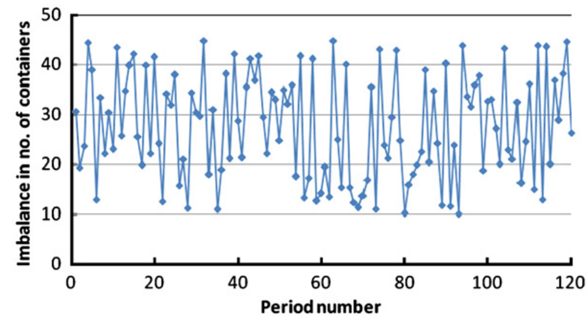

Fig. 9. Imbalance of the number of containers.

Table 2
Performance of the stacking algorithms on rehandle percentage.

	RSA (%)	VSA (%)	HSSA (%)
Average	44.99	26.16	18.53
Maximum	54.29	31.45	23.71
Minimum	40.00	22.18	14.50

6.4. Performance on location assignment

The performance of the proposed HSSA is compared with two other stacking algorithms: random stacking algorithm and vertical stacking algorithm.

(1) Random stacking algorithm (RSA): upon the arrival of an outbound container, the algorithm uses random search to find a yard bay that is not full in the pre-assigned block. If the bay consists of containers of a different destination, then the container cannot be stacked in this bay. Otherwise, the container is then stacked in a randomly chosen slot in the bay. If no such bay can be found, the algorithm then determines a new empty bay where the container can be stacked.
(2) Vertical stacking algorithm (VSA) is the same with the HSSA when determining the bay for a container to stack. For selecting a specific slot in a certain bay for a container, the algorithm uses vertical stacking method described in Section 5.2.

To evaluate the performance of the different stacking algorithms on the handling efficiency of the loading operation, the total number of rehandle is calculated given a loading sequence. The number of rehandle for a target container is calculated as the number of containers located above this container at the time of loading. We measure the total number of rehandle as a percentage of the number of containers in the loading sequence. Table 2 compares performance of the three stacking algorithms on the average percentage of rehandle for different ships over all the planning horizons.

It is observed from the results that the HSSA is better than both the RSA and VSA in terms of reducing the percentage of rehandle. The average rehandle percentages for RSA and VSA are 44.99% and 26.16%, respectively. The average rehandle percentage for HSSA is 18.53%, which represents a large improvement over the average figure of 30% in the current real operation. These results demonstrate the effectiveness of the HSSA.

7. Conclusions

The storage location assignment problem for outbound containers in a maritime terminal is directly related to the handling efficiency of the loading operations and is difficult to solve due to the random arrival of the outbound containers. In this study, the problem is decomposed into two stages. The yard bays and the amount of locations in each yard bay that will be assigned to the containers bounded for different ships are determined in the first stage. A mixed integer programming model is formulated to solve the problem. The exact storage location for each container is determined in the second stage. The selection of the best location for a container in a given bay with an arbitrary configuration is described. Based on this, the hybrid sequence stacking algorithm is developed to determine the storage location for each container upon its arrival at the terminal. We considered a complex situation in which containers of different weights are mixed in the yard bays. This is consistent with the practice in Shanghai.

Experimental study shows the effectiveness of the proposed approach in both levels. Performance on total travel distance between the yard and the berth, the workload imbalance among different container blocks, and the percentage of rehandles in the loading operations are compared and analyzed.

An important extension of this research would be to consider the storage allocation for outbound containers and the loading sequence at the same time. Integrating these two decision making processes will determine both optimal locations and the corresponding handling sequence.

Acknowledgement

This work is supported by National Natural Science Foundation of China (Project number: 70802040, 70771065), the National High Technology Research and Development Program of China (863 Program, No. 2009AA043000), and the Science and Technology Commission of Shanghai Municipality (Project number: 08DZ2210102)

References

Castilho, B., Daganzo, C.F., 1992. Handling strategies for import containers at marine terminals. Transportation Research Part B 27, 151–166.
Dekker, R., Voogd, P., van Asperen, E., 2006. Advanced methods for container stacking. OR Spectrum 28, 563–586.
Hao, J., Ji, Z., Lin, Y., 2000. Study of optimization of a bay of stacking. Journal of Dalian University of Technology 40 (1), 102–105.
Kim, K.H., Kim, H.B., 1999. Segregating space allocation models for container inventories in port container terminals. International Journal of Production Economics 59, 415–423.
Kim, K.H., Kim, K.Y., 2007. Optimal price schedules for storage of inbound containers. Transport Research Part B 41, 892–905.
Kim, K.H., Park, K.T., 2003. A note on a dynamic space allocation method for outbound containers. European Journal of Operational Research 148 (1), 92–101.
Kim, K.H., Park, Y.M., Ryu, K.R., 2000. Deriving decision rules to locate export containers in container yards. European Journal of Operational Research 124 (1), 89–101.
Preston, P., Kozan, E., 2001. An approach to determine storage locations of containers at seaport terminals. Computers & Operations Research 28, 983–995.
Taleb-Ibrahimi, M., Castilho, B., Daganzo, C.F., 1993. Storage space vs. handling work in container terminals. Transportation Research Part B 27 (4), 13–32.
Zhang, C., Liu, J., Wan, Y.-W., Murty, K.G, Linn, R.J., 2003. Storage space allocation in container terminals. Transportation Research Part B 37, 883–903.