

Research Article

Storage Space Allocation of Inbound Container in Railway Container Terminal

Li Wang, Xiaoning Zhu, and Zhengyu Xie

School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China

Correspondence should be addressed to Xiaoning Zhu; xnzhu@bjtu.edu.cn

Received 21 February 2014; Revised 14 June 2014; Accepted 29 June 2014; Published 22 July 2014

Academic Editor: Hu Shao

Copyright © 2014 Li Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Efficient storage strategy of railway container terminals is important in balancing resource utilization, reducing waiting time, and improving handling efficiency. In this paper, we consider the formulation and solution algorithm for storage space allocation problem of inbound containers in railway container terminal. The problem is formulated as two-stage optimization models, whose objectives are balancing the workload of inbound containers and reducing the overlapping amounts. An algorithm implement process based on rolling horizon approach is designed to solve the proposed models. Computational experiments on an actual railway container terminal show that the proposed approach is effective to solve space allocation problem of inbound container and is significant for the operation and organization of railway container terminals.

1. Introduction

With rapid development of passenger-dedicated railway in China, more transportation capacities of railway freight corridor are released, which puts forward huge opportunities and challenges for railway container transportation. At present, operation and organization of railway container terminals cannot meet developing demands of container transportation. To enable container to rapidly transfer between rail and truck, modern transshipment technologies are required to improve the resource utilization in railway container terminals.

As key resources of railway container terminals, storage spaces are responsible for temporarily storing containers which are moved over long distances by container trains and short distances by trucks. Storage space allocation is important in balancing space utilization, improving the efficiency of container handling, and reducing the turnaround time of containers. The storage space allocation is a vital basis and constraint for other resources utilization in railway container terminal.

Inbound and outbound container operations in railway container terminal are different. Inbound containers arrive predictably in large quantities, are temporarily stored in container yard, and depart one by one in an unpredictable

sequence. On the contrary, outbound containers arrive in a random sequence and depart predictably. This paper focuses on the inbound containers in railway container terminals.

The rest of this paper is organized as follows: the relevant literature is reviewed in the next section. The storage space allocation problem of inbound containers in railway container terminals is described in Section 3 and formulated in Section 4. An algorithm implement process based on rolling horizon approach is developed in Section 5. Computational results are reported in Section 6 and finally Section 7 covers the conclusion.

2. Literature Review

The storage space allocation problem of inbound containers in railway container terminal belongs to the storage space allocation problem (SSAP) which is defined as the temporary allocation of the inbound/outbound containers to the storage blocks at each time period with aim of balancing the workload between blocks in order to minimize the storage/retrieval times of containers [1].

The SSAP was firstly formulated for a container terminal in Hong Kong [2]. The problem was solved by a rolling horizon approach. In each planning horizon, the problem

is decomposed into two levels and formulated each level as a mathematical programming model. In order to determine the storage location of arriving export containers by considering its weight, a dynamic programming model is formulated to minimize the number of relocation movements expected for the loading operation [3]. The process of determining the storage locations for outbound containers was divided into two stages: space allocation stage and stage of locating individual containers [4]. A storage location assignment problem for outbound containers of maritime terminal was decomposed into two stages. 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 [5].

An approach for allocating storage space to groups of outbound containers in port container terminals was proposed with considering the impacts of various space-reservation strategies on the productivity of the loading operation [6]. For finding the best allocation of containers in a yard bay in order to minimize the number of reshuffles, a domain-dependent heuristically guided planner for obtaining the optimized reshuffling plan was proposed with a stacking state and a container demand known [7]. In order to improve the operations efficiency of retrieving inbound containers in a modern automatic container terminal, inbound container space allocation models were proposed to optimally allocate the arrival inbound containers so as to minimize the expected container retrieval time [8]. A novel approach using ant-based control was proposed for allocating containers to storage blocks in a marine container terminal with the competing objectives of balancing the workload among yard blocks and minimizing the distance traveled of trucks between yard blocks and berths [9].

In order to solve the storage space problem of outbound containers for utilizing space efficiently and make loading operations more efficient, two heuristic algorithms are suggested based on the duration-of-stay of containers and the subgradient optimization technique, respectively [4]. An extended version of SSAP was proposed and an efficient genetic algorithm was developed to solve the extended problem in a container terminal [1]. Construction algorithms and a tabu search heuristic are presented for dynamic space allocation problem to optimize the space/resource assignments during the implementation of project activities [10]. Three tabu search heuristics are presented for dynamic space allocation problem. The first heuristic is a simple basic tabu search heuristic. The second heuristic adds diversification and intensification strategies to the first and the third heuristic is a probabilistic tabu search heuristic [11].

A hybrid insertion algorithm is designed for solving the problem which integrates the yard truck scheduling and the storage allocation to minimize the weighted sum of total delay of requests and the total travel time of yard trucks [12]. For the problem of determining the stacking positions for incoming containers in automated container terminals, an online search algorithm was proposed to dynamically adjust and optimize a stacking policy by continuously generating and evaluating variants of stacking policies [13]. A decision support system was present to manage container

stacking problem, berth allocation problem, and the quay crane assignment problem in a coordinated way. A domain-oriented heuristic planner for calculating the number of reshuffles needed to allocate the containers in the appropriate place [14]. A construction and a hybrid algorithm (HGT) based on the GRASP and tabu search metaheuristics were proposed to solve the dynamic space allocation problem, where project duration is divided into a number of consecutive periods, each of them associated with a number of activities [15].

According to the literature review above, most studies focused on the storage space allocation problem in maritime container terminals. Specific literature on railway container terminal is scarce. Because storage strategy of inbound and outbound containers in railway container terminals is obviously different from maritime container terminals, existing studies are hardly applied in railway container terminals. In this paper, we consider the storage space allocation problem of inbound container in railway container terminals. Given that container arrival-departure time and operation sequences are known, two-stage storage space allocation models are proposed, whose objectives are to balance workloads among container blocks and assign containers to the optimum positions.

3. Problem Description

The Chinese railway container terminals have advanced arrival-departure lines, storage spaces, and handling equipment. Figure 1 gives a schematic representation of a typical railway container terminal in China. Our study is based on the configuration and layout of the representation in Figure 1.

As observed in Figure 1, container yard of railway container terminal is composed by inbound container yard, outbound container yard, and auxiliary container yard. Since most of inbound container allocation operations occur in inbound container yard, we set inbound container yard as the study scope of this paper.

According to the inbound container status in different handling stages, containers to be handled in inbound container yard can be classified into the following three types.

- (i) Inbound containers on container train wait for unloading and allocating to the inbound container yard, abbreviated as ICT.
- (ii) ICT containers temporarily stored in inbound container yard wait for loading to trucks for customers, abbreviated as ICTY.
- (iii) ICT containers are unloaded and directly loaded to trucks for customers, abbreviated as ICTT.

According to optimization objectives of SSAP, most literatures decomposed the storage space allocation problem into two stages. The first stage is to balance containers workloads among blocks and evenly allocate containers to each block. The second stage is the slot allocation for containers which are allocated to blocks based on the first stage optimization results. In this paper, our study decomposed the storage space allocation problem of inbound container in railway container

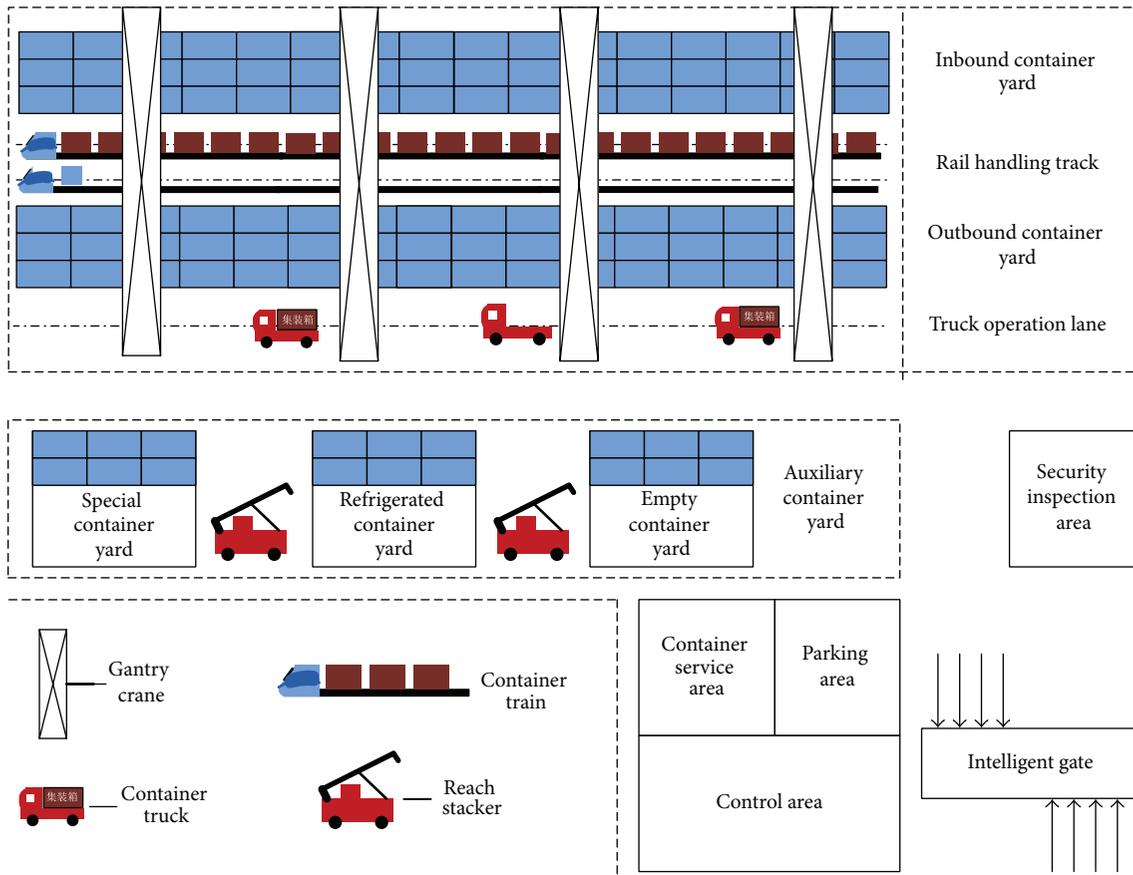


FIGURE 1: Schematic representation of a railway container terminal.

terminal into two stages based on the optimization objectives of SSAP.

- (i) The first stage is workload balance similar to the exciting study, whose objective is to balance workloads of inbound containers among blocks and evenly allocate inbound containers to each block.
- (ii) The second stage is container slot allocation based on the first stage optimization results, whose objective is to minimize overlapping amounts of ICT.

4. Problem Formulation

In this section, according to the problem description above, the storage space allocation problem of inbound container in railway container terminal is formulated as two-stage optimization models based on rolling horizon approach. At each planning epoch, we plan for a fixed horizon in immediate future and execute the plan accordingly up to the next planning epoch; then we formulate a new plan based on the latest information; this pattern goes on continually [2]. The rolling of planning horizon is shown in Figure 2. The workload balance is implemented in each planning epoch, and the container slot allocation is implemented in each period of planning epoch.

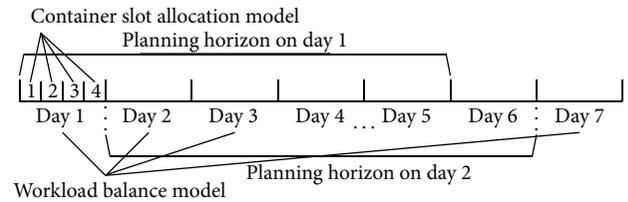


FIGURE 2: Rolling of planning horizon.

4.1. Assumptions. The following four assumptions are introduced for the formulation of the problem.

- (1) There is enough resource, that is, gantry crane and container yard space, to handle the allocation workload at the considered block.
- (2) The arriving and departure time of containers are known in advance and there is no time delay in scheduling period.
- (3) Handling sequence of containers is assumed to be known.
- (4) The containers in the model are assumed to be of one size.

4.2. Workload Balance Model

4.2.1. *Notations and Variables.* The notations and variables of workload balance model are defined as follows:

IB : total number of blocks in inbound container yard;

T : total number of planning periods in a planning epoch;

C_i : storage capacity of block i , $1 \leq i \leq IB$;

V_{i0} : initial inventory of block i , $1 \leq i \leq IB$;

V_{it} : total number of containers in block i at beginning of period t , $1 \leq i \leq IB$, $1 \leq t \leq T$;

$ICTY_{it}^0$: number of ICTY in block i that are picked up at beginning of period t , $1 \leq i \leq IB$, $1 \leq t \leq T$;

ICT_{tk} : number of ICT that are unloaded from trains in period t and to be picked up in period $t+k$, $1 \leq t \leq T$, $0 \leq k \leq T-t$;

ICT_{it} : number of ICT in block i that are unloaded from trains in period t , $1 \leq i \leq IB$, $1 \leq t \leq T$;

ICT_{itk} : number of ICT in block i that are unloaded from trains in period t and to be picked up in period $t+k$, $1 \leq i \leq IB$, $1 \leq t \leq T$, $0 \leq k \leq T-t$;

$ICTY_{it}$: number of ICTY in block i that are picked up in period t , $1 \leq i \leq IB$, $1 \leq t \leq T$;

$ICTT_t$: number of ICTT that are unloaded and directly loaded to trucks in period t , $1 \leq t \leq T$.

4.2.2. *Objective Function.* By attention to the problem description in Section 3, the objective function of workload balance model is written as follows:

$$\text{Min} \sum_{t=1}^T \left[\max_{\{i\}} (ICT_{it} + ICTY_{it}) - \min_{\{i\}} (ICT_{it} + ICTY_{it}) \right]. \quad (1)$$

Objective function is to balance workloads of inbound containers among blocks and evenly allocate ICT to blocks in each planning epoch.

4.2.3. *Constraints.* The constraints of workload balance model are introduced as follows to ensure the practical feasibility of the solution.

(1) Constraints on ICT:

$$ICT_{tk} = \sum_{i=1}^{IB} ICT_{itk}, \quad t = 1, 2, \dots, T; \quad k = 0, 1, \dots, T-t, \quad (2)$$

$$ICT_{it} = \sum_{k=0}^{T-t} ICT_{itk}, \quad i = 1, 2, \dots, IB; \quad t = 1, 2, \dots, T. \quad (3)$$

Constraint (2) ensures that the total number of ICT that are unloaded from trucks in period t and to be loaded onto rail vehicles in period $t+k$ is the sum of these containers assigned to all the blocks. Constraint

(3) ensures that the total number of ICT in block i that are unloaded from trucks in period t is the sum of these containers loaded onto rail vehicles in period $t+k$ in this block.

(2) Constraints on ICTY:

$$ICTY_{it} = ICTY_{it}^0 + \sum_{k=0}^{t-1} ICT_{i(t-k)k}, \quad (4)$$

$$i = 1, 2, \dots, IB; \quad t = 1, 2, \dots, T.$$

Constraint (4) indicates that the number of ICTY in block i during period t is the sum of initial number of ICTY in block i and the number of ICTY transferred from the ICT that unloaded in the block in planning epoch.

(3) Constraints on ICTT:

$$ICTT_t = \sum_{i=1}^{IB} (ICT_{it} - ICT_{itk}), \quad (5)$$

$$t = 1, 2, \dots, T, \quad k = t, t+1, \dots, T.$$

Constraint (5) indicates that the number of ICTY in block i during period t is the sum of initial number of ICTY in block i and the number of ICTY transferred from the ICT that unloaded in the block in planning epoch.

(4) Capacity constraints:

$$V_{it} = V_{i(t-1)} + (ICT_{it} - OICT_{it}), \quad (6)$$

$$i = 1, 2, \dots, IB; \quad t = 1, 2, \dots, T$$

$$V_{it} \leq C_i, \quad i = 1, 2, \dots, IB; \quad t = 1, 2, \dots, T. \quad (7)$$

Constraint (6) indicates the inventory of containers in block i at beginning of period t . Constraint (7) indicates the storage capacity of block i .

(5) Integer constraint:

All variables take up nonnegative integer values. (8)

4.3. Container Slot Allocation Model

4.3.1. *Notations and Variables.* The notations and variables of container slot allocation model are defined as follows:

N : total number of ICT which are allocated in considered block at the same planning period;

v : the sequence number of allocated container;

B : total number of bays in the considered block;

R : total number of rows in considered block;

L : maximum layer number of stack;

b : bay identifier of container slot, $1 \leq b \leq B$;

r : row identifier of container slot, $1 \leq r \leq R$;

- l : layer identifier of container slot, $1 \leq l \leq L$;
 $s(r, b, l)$: container slot of r row, b bay, and l layer;
 t_{rbl} : departure time of container in container slot of r row, b bay, and l layer;
 K_v^{ICT} : overlapping amounts of the v ICT;
 M : an infinitesimal number;
 S_{rbl} : if $s(r, b, l)$ has container, $S_{rbl} = 1$. Otherwise, $S_{rbl} = 0$;
 S_{vrbl} : if the v container is allocated to $s(r, b, l)$, $S_{vrbl} = 1$. Otherwise, $S_{vrbl} = 0$;
 $K_{rbl,rb(l-e)}^{\text{ICT}}$: overlapping of $s(r, b, l - e)$ after ICT was allocated to $s(r, b, l)$. If $t_{rbl} < t_{rb(l-e)}$, $K_{rbl,rb(l-e)}^{\text{ICT}} = 1$. Otherwise, $K_{rbl,rb(l-e)}^{\text{ICT}} = 0$;
 H_v : if there are no empty bays while the v ICT is allocated, $H_v = 1$. Otherwise, $H_v = 0$.

4.3.2. *Objective Functions.* By attention to the problem description in Section 3, the objective functions of container slot allocation model are written as follows:

$$\text{Min} \sum_{v=1}^D K_v^{\text{ICT}}. \quad (9)$$

The second stage objective function (9) minimizes overlapping amount which is caused by ICT allocated in considered block at same planning period.

4.3.3. *Constraints.* The constraints of container slot allocation model are introduced as follows to ensure the practical feasibility of the solution.

(1) Overlapping amounts constrains of ICT:

$$K_v^{\text{ICT}} = (1 - H_v) M + H_v \cdot \min_{e=1}^{l-1} K_{rbl,rb(l-e)}^{\text{ICT}}, \quad (10)$$

$$v = 1, 2, \dots, N, \quad r = 1, 2, \dots, R,$$

$$b = 1, 2, \dots, B, \quad l = 2, 3, \dots, L.$$

Constraint (10) represents the calculation of overlapping amount which is caused by ICT allocated in considered block.

(2) Allocation constrain:

$$S_{vrbl} - S_{rb(l-1)} \leq 0, \quad r = 1, 2, \dots, R, \quad (11)$$

$$b = 1, 2, \dots, B, \quad l = 2, 3, \dots, L, \quad v = 1, 2, \dots, N.$$

Constraint (11) ensures that each container cannot be allocated upon the empty container slot.

(3) Allocation preferences constraints:

$$\sum_{v=1}^N \sum_{r=1}^m S_{vrbl} - \sum_{v=1}^N \sum_{r=R}^{R-(m-1)} S_{vrbl} \leq 0, \quad (12)$$

$$b = 1, 2, \dots, B, \quad l = 1, 2, \dots, L, \quad m = \frac{R}{2} - 1.$$

Constraint (12) indicates the allocation preferences constraint of ICT. It ensures that the allocation positions of ICT are close to the truck operation lane in order to reduce loading time of ICTY.

5. Solution Algorithm

In order to solve the two-stage models presented above, an algorithm implement process based on rolling horizon approach is proposed in this section. The workload balance model is converted to a linear integer programming model and a heuristic algorithm is designed to solve the container slot allocation model. The implement process is shown in Figure 3.

5.1. *Workload Balance Model Conversion.* The workload balance model proposed in Section 4 is a nonlinear model, because of the objective function of model. In order to obtain a solution, a conversion method should be used to convert the model to a linear model. According to the conversion mentioned in [2], we define $A_t = \max_{\{i\}} (\text{ICT}_{it} + \text{ICTY}_{it})$ and $B_t = \min_{\{i\}} (\text{ICT}_{it} + \text{ICTY}_{it})$. Then the workload balance model can be converted as the linear integer programming model as follows:

$$\text{Min} \sum_{t=1}^T (A_t - B_t). \quad (13)$$

The constraints include (2) to (8) and the constraints on A_t and B_t :

$$\text{ICT}_{it} + \text{ICTY}_{it} \leq A_t, \quad i = 1, 2, \dots, IB; \quad t = 1, 2, \dots, T \quad (14)$$

$$\text{ICT}_{it} + \text{ICTY}_{it} \geq B_t, \quad i = 1, 2, \dots, IB; \quad t = 1, 2, \dots, T.$$

After the conversion of workload balance model, it can be solved by Lingo.

5.2. *HA Implementation for Container Slot Allocation Model.* The notations of HA are shown in Table 1 and the procedure of HA is shown as follows.

Step 1. According to the initial block information at the beginning of period, get a feasible allocated set of container slots F by removing infeasible container slots. Parameter initialization: let $v = 1$, $r = R$, $b = 1$, $K = \{\phi\}$, $S = \{\phi\}$, $K_v^{\text{ICT}} = I$, and go to Step 2.

Step 2. Allocate the v ICT. Search empty container slots in block row from R to $R - (m - 1)$. If empty container slot exists, go to Step 3. If empty container slot does not exist, go to Step 4.

Step 3. Let $S = S \cup \{S_{rbl}^v\}$ and $v = v + 1$; then if $v \leq N$, go to Step 2. Otherwise, let $K_v^{\text{ICT}} = 0$ and go to Step 8.

Step 4. If $s(r, b, l) \notin F$, go to Step 6. Otherwise allocate container to $s(r, b, l)$ and judge the feasibility of solution. If the solution is not feasible, go to Step 8. If the solution is feasible, go to Step 5.

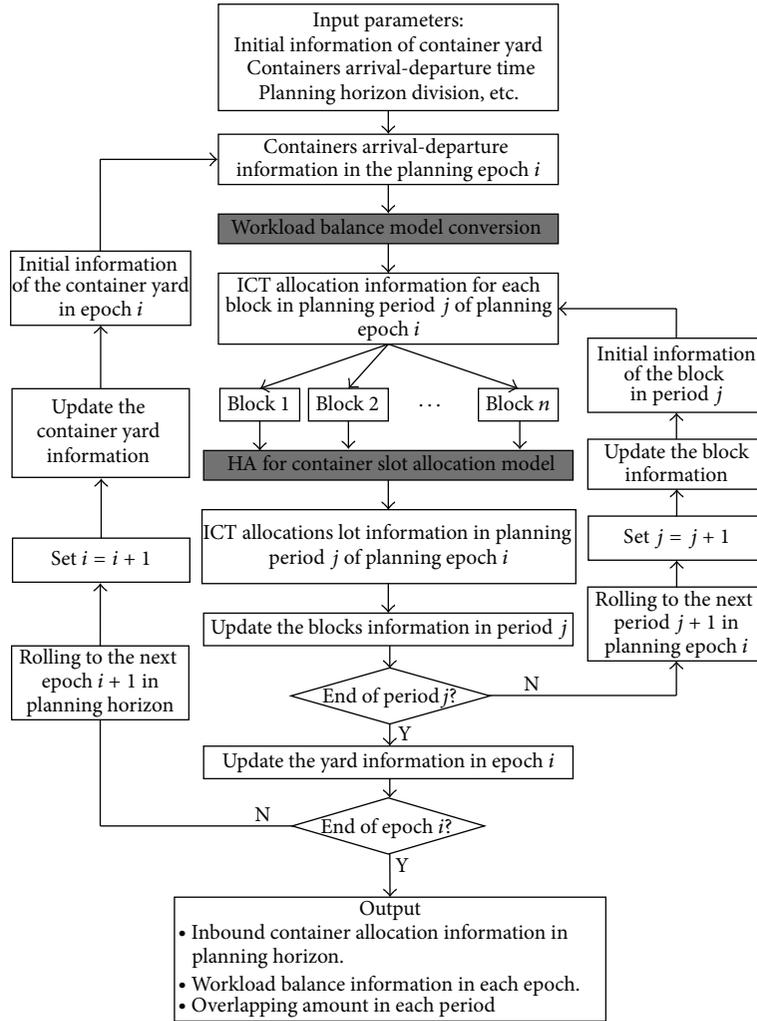


FIGURE 3: Algorithm implement process.

TABLE 1: Notations.

Notation	Declaration
K	The set of overlapping amounts
S	The optimal set of slots allocation with minimum overlapping amount
S_{rbl}^v	The container allocated to $s(r, b, l)$
K_n^{ICT}	the overlapping amount increased by the n feasible solution of ICT
d_{vrb}^{\min}	The minimum crane operation distances of the v container allocated to $s(r, b, l)$
A	The alternative set of the v allocated slot with minimum overlapping amount
F	The feasible allocated set of container slots
n	The number of feasible solutions
d_{vrb}	The operation distance of crane for the v allocated container
I	An arbitrary positive big number

Step 5. Let the solution be K_n^{ICT} and compare K_n^{ICT} and K_v^{ICT} . If $K_n^{\text{ICT}} > K_v^{\text{ICT}}$, go to Step 6. Otherwise, if $K_n^{\text{ICT}} = K_v^{\text{ICT}}$, let

$A = A \cup \{S_{rbl}^v\}$ and go to Step 6; otherwise, let $K_v^{\text{ICT}} = K_n^{\text{ICT}}$, $A = \{\phi\}$, $A = A \cup \{S_{rbl}^v\}$, and go to Step 6.

Step 6. Let $b = b + 1$. If $b \leq B$, go to Step 4. Otherwise, let $r = r - 1$. If $r \geq 1$, go to Step 4. If $r < 1$, calculate the d_{vrb} in set A and select d_{vrb}^{\min} , set $K = K \cup \{K_v^{\text{ICT}}\}$, $S = S \cup \{S_{rbl}^v\}$, and go to Step 7.

Step 7. Allocation of the v ICT has finished. Let $r = R$, $b = 1$, and $v = v + 1$; then if $v \leq N$, go to Step 2; otherwise go to Step 8.

Step 8. Calculate the overlapping amounts based on K and output the set of container slot allocation S . Procedure terminates.

6. Computational Experiment

To illustrate the proposed model and algorithm for space allocation problem of inbound container in railway container terminal, computational experiments are performed by using the actual data from a specific railway container terminal

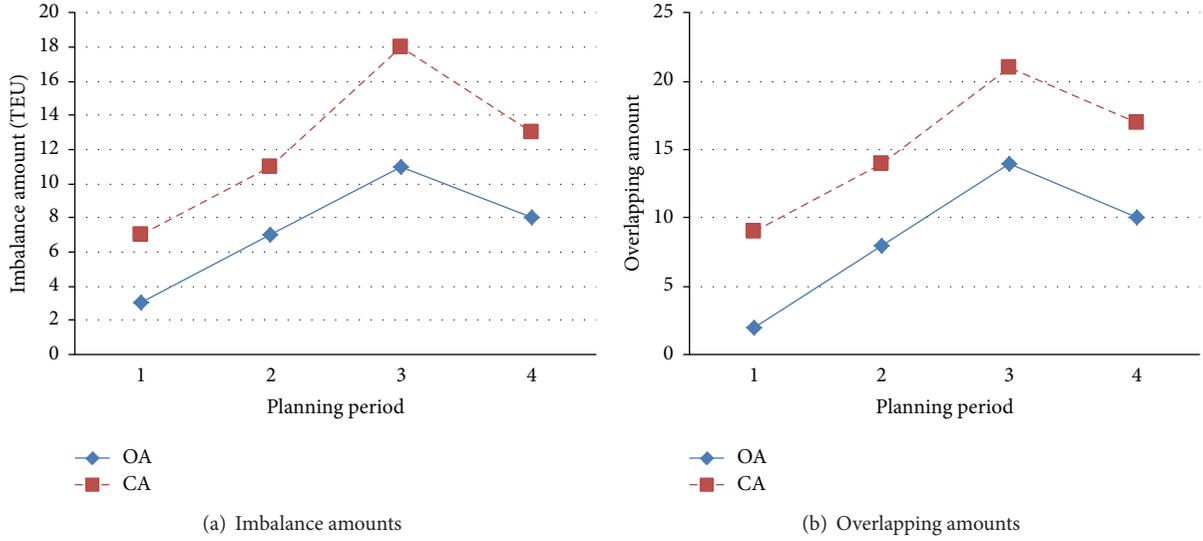


FIGURE 4: Computational results of 1 day.

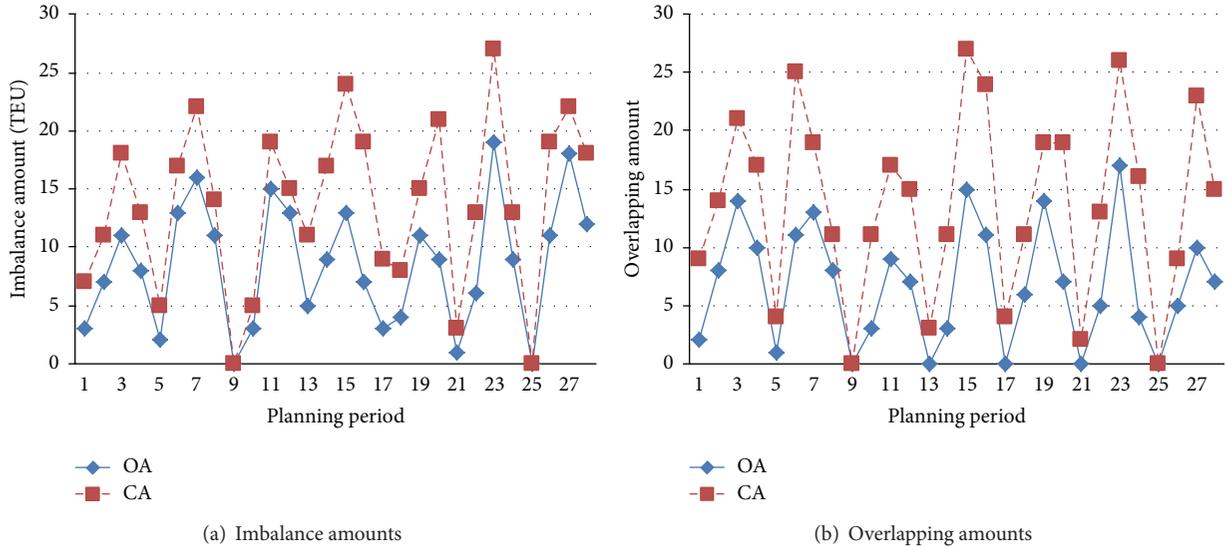


FIGURE 5: Computational results of 7 days.

in China [16]. In order to evaluate the improvement of our approach, a comparison is made between our approach and random allocation algorithm which is currently used in railway container terminals. Furthermore, to evaluate the effectiveness and practicability of our approach, numerical experiments of 7 days and 30 days are carried out. To implement the proposed algorithm, the parameters related to the specific railway container terminal are needed. There are four blocks in the inbound container yard. Each block is composed by 30 bays, 6 rows, and 2 layers. The numerical experiments are performed based on a personal computer with Intel Core i5-2450M @ 2.50 GHz processors and 4 GB RAM.

Because most of ICTY are picked up no more than two days after they allocated to blocks, we choose 3 days as a planning horizon, 1 day as a planning epoch, and 6

TABLE 2: Inbound container information of 4 planning periods in 1 day (TEU).

Container type	$t = 1$	$t = 2$	$t = 3$	$t = 4$
ICT	20	25	135	40
ICTY	15	65	79	67
ICTT	2	3	7	3

hours as a planning period. There are 4 planning periods in one planning epoch and 12 planning periods in one planning horizon. A small size sample of 1 day is carried out firstly. The inbound container information of 4 planning periods in 1 day is shown in Table 2. The ICTY loading plan of each block in planning period is shown in Table 3.

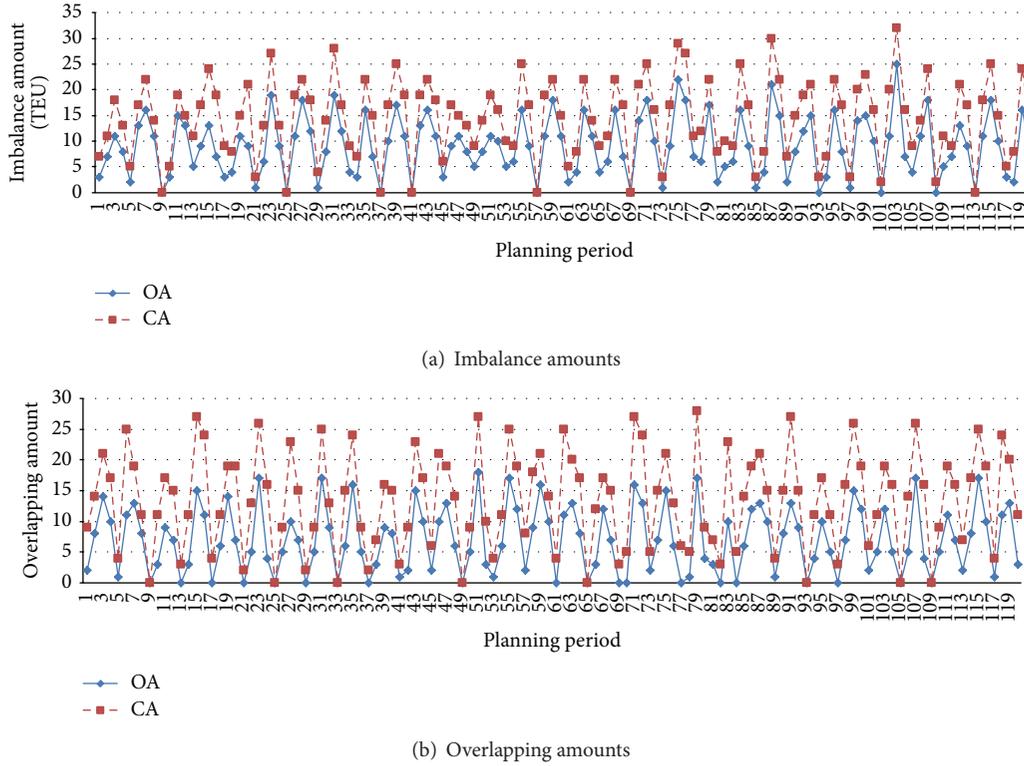


FIGURE 6: Computational results of 30 days.

TABLE 3: ICTY loading plan of each block in planning period (TEU).

Container type	$i = 1$				$i = 2$				$i = 3$				$i = 4$			
	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 1$	$t = 2$	$t = 3$	$t = 4$
ICTY	4	0	6	5	18	27	13	7	13	22	16	28	27	19	12	9

TABLE 4: Comparison between OA and RAA in 1 day.

Planning period	Our approach (OA)		Random allocation algorithm (RAA)		GAP ₁	GAP ₂
	Imbalance amounts	Overlapping amounts	Imbalance amounts	Overlapping amounts		
1	3	2	7	9	57.1%	77.8%
2	7	8	11	14	36.3%	42.9%
3	11	14	18	21	38.8%	33.3%
4	8	10	13	17	38.4%	41.2%

Notes: $GAP_1 = (\text{imbalance amounts obtained from RAA} - \text{imbalance amounts obtained from OA}) * 100 / \text{imbalance amounts obtained from RAA}$.
 $GAP_2 = (\text{overlapping amounts obtained from RAA} - \text{overlapping amounts obtained from OA}) * 100 / \text{overlapping amounts obtained from RAA}$.

Based on the computational example above, experiment is conducted, and a comparison between our approach (OA) and random allocation algorithm (RAA) is made to evaluate the performance of OA on space allocation problem of inbound container in railway container terminal. The computational results are shown in Table 4 and Figure 4.

As observed in Table 4 and Figure 4, the imbalance amounts and overlapping amounts obtained by OA are both fewer than the amounts obtained by CA, the average GAP of imbalance amounts is 42.7%, and the average GAP of

overlapping amounts is 48.8%. In order to evaluate the effectiveness and practicability of our approach, numerical experiments of 7 days and 30 days are conducted. The computational results of 7 days are shown in Table 5 and Figure 5 and of 30 days are shown in Figure 6.

As observed in Table 5 and Figures 5 and 6, the performance of our approach is satisfactory in solving different size instances. The results of computational experiments indicate that our approach is efficient to solve space allocation problem of inbound container in railway container terminals.

TABLE 5: Comparison between OA and RAA in 7 days.

Planning period	Our approach (OA)		Random allocation algorithm (RAA)		GAP ₁	GAP ₂
	Imbalance amounts	Overlapping amounts	Imbalance amounts	Overlapping amounts		
1	3	2	7	9	57.1%	77.8%
2	7	8	11	14	36.4%	42.9%
3	11	14	18	21	38.9%	33.3%
4	8	10	13	17	38.5%	41.2%
5	2	1	5	4	60.0%	75.0%
6	13	11	17	25	23.5%	56.0%
7	16	13	22	19	27.3%	31.6%
8	11	8	14	11	21.4%	27.3%
9	0	0	0	0	0.0%	0.0%
10	3	3	5	11	40.0%	72.7%
11	15	9	19	17	21.1%	47.1%
12	13	7	15	15	13.3%	53.3%
13	5	0	11	3	54.5%	100.0%
14	9	3	17	11	47.1%	72.7%
15	13	15	24	27	45.8%	44.4%
16	7	11	19	24	63.2%	54.2%
17	3	0	9	4	66.7%	100.0%
18	4	6	8	11	50.0%	45.5%
19	11	14	15	19	26.7%	26.3%
20	9	7	21	19	57.1%	63.2%
21	1	0	3	2	66.7%	100.0%
22	6	5	13	13	53.8%	61.5%
23	19	17	27	26	29.6%	34.6%
24	9	4	13	16	30.8%	75.0%
25	0	0	0	0	0.0%	0.0%
26	11	5	19	9	42.1%	44.4%
27	18	10	22	23	18.2%	56.5%
28	12	7	18	15	33.3%	53.3%

Notes: $GAP_1 = (\text{imbalance amounts obtained from RAA} - \text{imbalance amounts obtained from OA}) * 100 / \text{imbalance amounts obtained from RAA}$.
 $GAP_2 = (\text{overlapping amounts obtained from RAA} - \text{overlapping amounts obtained from OA}) * 100 / \text{overlapping amounts obtained from RAA}$.

7. Conclusion

In this paper, we considered the space allocation problem of inbound container in railway container terminals with the arrival-departure time and operation sequence of containers known. Two-stage optimization models were proposed; the first stage is workload balance model, whose objective is to balance workloads of inbound containers among blocks and evenly allocate inbound containers to each block. The second stage is container slot allocation model, whose objective is to minimize overlapping amounts of ICT. An algorithm implement process based on rolling horizon approach is designed to solve the proposed models. Computational experiments on an actual railway container terminal show that the models and algorithm proposed in this paper are effective to balance workloads of inbound containers and reduce the overlapping amounts.

In future, proposing a stochastic programming model for container slot allocating problem by considering random factors is a possibility for further research.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant no. 60870014), the Specialized Research Fund for the Doctoral Program of Higher Education (Grant no. 20130009110001), and the Major Research plan of the National Natural Science Foundation of China (Grant no. 71390332).

References

- [1] M. Bazzazi, N. Safaei, and N. Javadian, "A genetic algorithm to solve the storage space allocation problem in a container terminal," *Computers and Industrial Engineering*, vol. 56, no. 1, pp. 44–52, 2009.

- [2] C. Zhang, J. Liu, Y. Wan, K. G. Murty, and R. J. Linn, "Storage space allocation in container terminals," *Transportation Research B: Methodological*, vol. 37, no. 10, pp. 883–903, 2003.
- [3] K. H. Kim, Y. M. Park, and K. Ryu, "Deriving decision rules to locate export containers in container yards," *European Journal of Operational Research*, vol. 124, no. 1, pp. 89–101, 2000.
- [4] K. H. Kim and K. T. Park, "A note on a dynamic space-allocation method for outbound containers," *European Journal of Operational Research*, vol. 148, no. 1, pp. 92–101, 2003.
- [5] L. Chen and Z. Lu, "The storage location assignment problem for outbound containers in a maritime terminal," *International Journal of Production Economics*, vol. 135, no. 1, pp. 73–80, 2012.
- [6] Y. J. Woo and K. H. Kim, "Estimating the space requirement for outbound container inventories in port container terminals," *International Journal of Production Economics*, vol. 133, no. 1, pp. 293–301, 2011.
- [7] M. Rodriguez-Molins, M. A. Salido, and F. Barber, "Intelligent planning for allocating containers in maritime terminals," *Expert Systems with Applications*, vol. 39, no. 1, pp. 978–989, 2012.
- [8] M. Yu and X. Qi, "Storage space allocation models for inbound containers in an automatic container terminal," *European Journal of Operational Research*, vol. 226, no. 1, pp. 32–45, 2013.
- [9] O. Sharif and N. Huynh, "Storage space allocation at marine container terminals using ant-based control," *Expert Systems with Applications*, vol. 40, no. 6, pp. 2323–2330, 2013.
- [10] A. R. McKendall Jr. and J. R. Jaramillo, "A tabu search heuristic for the dynamic space allocation problem," *Computers and Operations Research*, vol. 33, no. 3, pp. 768–789, 2006.
- [11] A. R. McKendall Jr., "Improved Tabu search heuristics for the dynamic space allocation problem," *Computers and Operations Research*, vol. 35, no. 10, pp. 3347–3359, 2008.
- [12] D. Lee, J. X. Cao, Q. Shi, and J. H. Chen, "A heuristic algorithm for yard truck scheduling and storage allocation problems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 45, no. 5, pp. 810–820, 2009.
- [13] T. Park, R. Choe, Y. Hun Kim, and K. Ryel Ryu, "Dynamic adjustment of container stacking policy in an automated container terminal," *International Journal of Production Economics*, vol. 133, no. 1, pp. 385–392, 2011.
- [14] M. A. Salido, M. Rodriguez-Molins, and F. Barber, "A decision support system for managing combinatorial problems in container terminals," *Knowledge-Based Systems*, vol. 29, pp. 63–74, 2012.
- [15] G. C. da Silva, L. Bahiense, L. Satoru Ochi, and P. O. Boaventura-Netto, "The dynamic space allocation problem: Applying hybrid GRASP and Tabu search metaheuristics," *Computers and Operations Research*, vol. 39, no. 3, pp. 671–677, 2012.
- [16] L. Wang, *Key resources scheduling optimization theory and method of railway container terminal [Ph.D. thesis]*, Beijing Jiaotong University, Beijing, China, 2014, (Chinese).



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

