Applied Mathematical Modelling 77 (2020) 1000–1024

Contents lists available at ScienceDirect

Applied Mathematical Modelling

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

Two-stage search algorithm for the inbound container unloading and stacking problem

Huiling Zhu [a], Mingjun Ji [b,*], Wenwen Guo [b]

[a] School of Maritime Economics and Management, Dalian Maritime University, Dalian 116026, China
[b] School of Transportation Engineering, Dalian Maritime University, Room 503 of Management building A, Linghai Road No.1, Dalian, Liaoning 116026, China

ARTICLE INFO

Article history:
Received 12 October 2018
Revised 22 July 2019
Accepted 19 August 2019
Available online 31 August 2019

Keywords:
Transportation
Container rehandling
Unloading sequence
Yard stacking distribution
Two-stage search algorithm

ABSTRACT

This study focuses on the inbound container unloading and stacking problem at container terminals and achieves both a reasonable unloading sequence and the optimal yard stacking distribution. A formulation is proposed as the relational expression between the expected number of rehandles and the stacking height. Based on the formulation, an integer programming model is established to both find the optimal stacking distribution and unloading sequence and attempt to minimize the expected number of rehandles. The model can be solved by the commercial solver for small-scale instances. To solve for large-scale instances in the real world, a two-stage search algorithm is designed, therein incorporating an initial stage for generating the feasible solution and a neighborhood search stage for finding the optimal solution. The algorithm can find an optimal solution in polynomial time, which is proved by theoretical methods and evidenced by numerical experiments.

1. Introduction

Containerized sea-freight transportation has grown dramatically over the last two decades, much faster than other sea transportation modes [1]. At container terminals, containers are temporarily stored in an area called the stacking yard, where the operations of loading, unloading and premarshalling are performed to deal with the outbound and inbound containers [2]. A high throughput is expected from terminals because they load or unload up to approximately 10,000 containers per visit [3]. Thus, major maritime container terminals have been striving to improve the efficiency of container terminals and maintain their competitiveness by avoiding additional work in the loading and unloading operations [4].

Rashidi and Tsang [5] focused on scheduling problems, including berth allocation, storage space assignment, crane deployment, scheduling and routing of vehicles, and appointment times to external trucks. Stahlbock and Voss [6] classified the main operations at container terminals into berth allocation, stowage planning, crane scheduling, terminal transport optimization, and storage and stacking logistics. The operations in container terminals involve both macro and micro levels. In recent years, publications concerning container storage and stacking problem, especially container rehandling involved in the problem, have explored methods to increase operational efficiency in container terminals. Kim [7] proposed a methodology to estimate the expected number of rehandles necessary to pick up an arbitrary container and the total number of rehandles necessary to pick up all the containers in a bay, where the height and width were considered as important decision

* Corresponding author.
E-mail address: jmj@dlmu.edu.cn (M. Ji).

https://doi.org/10.1016/j.apm.2019.08.019
0307-904X/

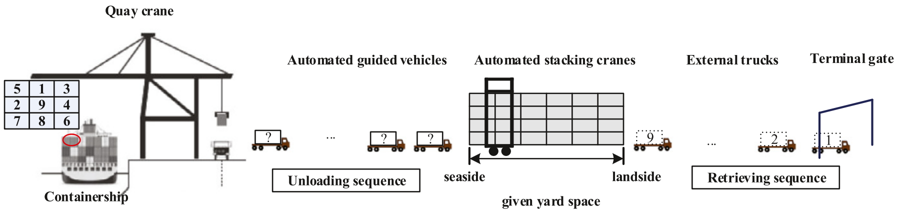

Fig. 1. An illustration of the ICUSP.

variables in designing the stacking distribution. Kim et al. [8] determined the storage location of an arriving export container considering its weight to minimize the number of relocation movements expected for the loading operation. Similar to the previous work, Imai et al. [9] developed mathematical models and solution methods to determine the overall number of rehandles.

The block relocation problem (BRP) focuses on the rehandling optimization when retrieving containers by a given sequence from the yard space. This was first studied by Kim and Hong [10]. Forster and Bortfeldt [11] presented a heuristic tree search procedure for the CRP. After surveying the work done on the BRP, Petering and Hussein [12] considered the containers in a single yard-bay, introduced a new mathematical formulation for the BRP, and expressed a new look-ahead algorithm to obtain superior solutions for the problem. Jin et al. [13] extended the previous work [14] and presented an improved greedy look-ahead heuristic to find an optimized operation plan for a crane with the fewest container relocations. Lee and Lee [15] extended the BRP to multiple bays by minimizing the number of container movements as well as the crane's working time. For single-bay scenarios, the problem is identical to the BRP.

The container pre-marshalling problem (CPMP) reallocates the containers in a given yard space, generating a new yard stacking distribution before the containers depart from the yard. Lee and Hsu [16] developed an IP model to address the CPMP. Caserta and Voss [17] provided a greedy heuristic based on the paradigm of corridor and roulette wheels. Lee and Chao [18] regarded complete solutions as search units obtained through a neighborhood search. Expósito-Izquierdo et al. [19] proposed a multi-start heuristic to minimize the number of rehandles, which is more efficient compared with Lee and Chao [18]. Bortfeldt and Forster [20] thoroughly described a tree search procedure and a new lower bound for the CPMP. Huang and Lin [21] and Wang et al. [22] solved the variants of the CPMP using heuristic algorithms.

The decisions concerning the rehandling operations in the BRP (CRP) or CPMP are made based on a given stacking distribution, studying the rehandling strategy in the retrieval process. Actually, the initial yard stacking distribution has a significant influence on the rehandling, and the unloading plan has an obvious effect on the yard stacking distribution. This study focuses on the mutual relationship among unloading sequence, yard stacking distribution and rehandling.

If the distribution of containers is well aligned in the space allocation, rehandling operations can be avoided. Therefore, the space allocation problem (SAP) is established to reduce the number of rehandles. Kim and Kim [23] suggested a mathematical model to allocate space for import containers in a way that minimizes the number of rehandles. Guldogan [24] and Park et al. [25] first assigned containers to blocks and second selected specific locations within the selected block according to the departure dates or weight. Woo and Kim [26] introduced a method for allocating storage space to groups of outbound containers and discussed the effects of various space-reservation strategies on the productivity of loading operations for outbound containers. Yu and Qi [27] designed an optimized block space allocation to store the inbound containers after they are unloaded from ships. Ng et al. [28] studied the problem of designing a yard template that balances the workload in an export yard. As an extension to the previous modeling work of Kim et al. [8] and Zhang et al. [29], Zhang et al. [30] further examined location assignment for outbound containers at container terminals. For the existed literatures about deterministic storage of inbound containers, the unloading plan is not selected as the constraint for space allocation problem.

In summary, both the BRP and the CPMP focus on the rehandling strategy in the retrieval process based on the given yard stacking distribution, and the SAP focuses on the stacking distribution not considering the unloading sequence. Generally, for the incoming containership, the SAP is used to reserve the storage apace and design the stacking strategy, which is regarded as the 1st stage. After that, containers are unloaded from the containership to the reserved space by the designed stacking strategy. To improve the handling efficiency, the CPMP focuses on reorganizing the stacking distribution in the idle time of handling equipment, which is regarded as the 2nd stage. Finally the containers would be retrieved from the storage space by a deterministic truck arrival sequence, which is the 3rd stage, defined as the BRP.

However, our study focuses on defining and formulating an optimization problem for the inbound containers unloading and stacking problem (ICUSP), considering the containership stowage and yard stacking distribution simultaneously as illustrated in Fig. 1. Referring to Stahlbock and Voss [6], the ICUSP belongs to the field of storage and stacking logistics based on berth allocation and stowage planning. This study focuses on the stacking strategy making, not considering the equipment scheduling. Inbound containers, unloaded from the containership and stacked in the yard for temporary storage, would be retrieved by the external trucks. An improper yard stacking distribution will result in lower space utilization and longer operation times at container terminals. As containers are stacked in the vertical direction, additional work, called container rehandling, is required if a container to be retrieved is not on the top tier of the yard-stack. For each yard-stack, if the container retrieval sequence from the yard coincides with the container storage order from the highest tier to the lowest tier of

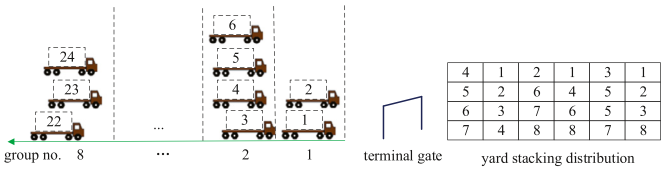

Fig. 2. An illustration of the ICUSP with an indefinite truck arrival time for the second scenario.

the yard space, rehandling can be completely eliminated. Actually, the container retrieval sequence is the same as the truck arrival sequence, but the truck arrival sequence rarely matches the container storage order. Thus, rehandling is unproductive but unavoidable. To minimize the number of rehandles, it is critical to match the container storage order to the truck arrival sequence. Thus, to minimize the number of rehandles in the yard, our work integrates the unloading sequence and the yard stacking distribution, in contrast to the BRP (CRP), the CPMP and the SAP. The ICUSP focuses on the operational level of the SAP, and it is the pre-processing of the container relocation problem. To sum up, the proposed ICUSP moves the operation planning developing forward, achieving multi-stage decision making in container terminals. The mutual influence of the unloading sequence and the stacking distribution on the number of rehandles makes the ICUSP more challenging.

The remainder of this study is organized as follows: Section 2 makes the problem description about the ICUSP from the particular scenario to the general scenario, and proposes a formulation to estimate the number of rehandles. Section 3 establishes a mathematical model for the ICUSP, which can be solved by the commercial solverfor small-scale instances. Section 4 designs a two-stage search algorithm for the ICUSP. Section 5 reports the results of the numerical experiments and verifies the efficiency of the algorithm. Finally, the conclusions are drawn in Section 6.

2. Problem description of the ICUSP

2.1. Property analysis of the ICUSP

According to Carlo et al. [31] and Borgman et al. [32], the containers stacking problems can be divided into two types: storage of individual containers or storage of group of containers. The groups of containers are classified by their departure time, which is determined by the truck arrival time window. And the group number refers to the corresponding containers' retrieval priority. The frequency of each group in statistics refers to the number of trucks arriving at the yard over a time period, and the trucks arriving over a time period are assigned to the same group. There generates two scenarios: (1) if the frequency for each group is no more than one, each inbound container would be characterized by its retrieval priority; (2) if there exists a group where the frequency exceeds one, more than one external truck will arrive during the corresponding time period to retrieve containers, and the containers cannot be characterized by its retrieval priority.

Property. For the first scenario, when each container has a unique retrieval priority, and the number of ship-stacks is greater than the allowed height of the yard-stack, the optimal stacking distribution without rehandles always exists.

To avoid rehandles in the yard space, the handling equipment can unload containers from the top of the ship-stacks by increasing retrieval priority to store them in a yard-stack orderly.

Generally, the number of ship-stacks for a ship-bay is usually more than 10, and the number of the allowed height of the yard-stack usually does not exceed 5 tiers, which is set to 4 tiers in this study. Thus, the number of ship-stacks is greater than the allowed height of the yard-stack in practice. In this scenario, the containers in a yard-stack could be from different ship-stacks with unobstructed unloading from the ship and efficient storage to the yard-stack without rehandles of trucks' retrieval. It easily generates the optimal stacking distribution without rehandles.

For the second scenario, the containers in a group have the same retrieval priority. Due to vague arrival information of trucks, none of the containers can be prioritized from other containers with the same retrieval priority. As illustrated in Fig. 2, trucks 1 and 2, arriving at the yard during the first time period, are classified as the first group and arrive prior to trucks 3, 4, 5 and 6, which are classified as the second group. The exact order of the trucks in each group is unknown. Thus, the truck arrival information can be determined in terms of groups instead of individuals. This is the focus of our study, as it is more common than the first scenario. When the length of each time period is set as short as possible, the containers can be classified individually, and the truck arrival sequence can be specified. When the length of each time period is longer, the frequency for each group increases, and each yard-stack has a greater probability of being loaded with containers with the same retrieval priority. Meanwhile, the number of rehandles may increase in this condition.

Without loss of generality, the movements are represented by {θ: [i, l], j}, in which θ is the identification number of the current unloaded container (here refers to the retrieval priority), i and l represent the identification number of ship-stack and ship-tier where container θ is originally located, and j represents the identification number of yard-stacks where container θ would be stacked. The unloading sequence can be {7: [3, 5], 1}, {4: [4, 5], 1}, {4: [5, 5], 1}, {3: [2, 5], 1}, {6: [6, 5], 2}, {5: [1, 5], 2}, {2: [4, 4], 2}, {1: [3, 4], 2}, {8: [6, 4], 3}, {8: [2, 4], 3}, {6: [5, 4], 3}, {1: [1, 4], 3}, {7: [4, 3], 4}, {7: [5, 3], 4}, {3: [2, 3], 4}, {2: [3, 3], 4}, {6: [3, 2], 5}, {5: [1, 3], 5}, {5: [6, 3], 5}, {4: [4, 2], 5}, {8: [3, 1], 6}, {3: [5, 2], 6}, {2: [4, 1], 6}, {1: [2, 2], 6}, and the generated yard stacking distribution is shown in Fig. 3(b). There are two containers with retrieval priority of "4" in the 1st stack, two containers with retrieval priority of "8" in the 3rd stack, two containers with retrieval priority of "7" in

5	3	7	4	4	6
1	8	1	2	6	8
5	3	2	7	7	5
	1	6	4	3	
		8	2		

(a)

3	1	1	2	4	1
4	2	6	3	5	2
4	5	8	7	5	3
7	6	8	7	6	8

(b)

4	1	2	1	3	1
5	2	6	4	5	2
6	3	7	6	5	3
7	4	8	8	7	8

(c)

Fig. 3. A ship-bay distribution including containers with identical retrieval priorities and two feasible yard stacking distributions.

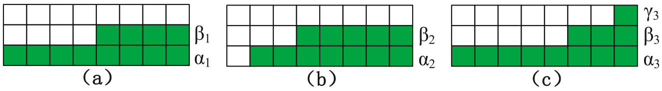

Fig. 4. Three yard stacking distributions of 12 containers.

the 4th stack, and two containers with retrieval priority of "5" in the 5th stack. In total, four rehandles, which represents the maximum number of rehandles, are produced pessimistically. The pessimistic rehandles occur when the lower container in a yard-stack is retrieved earlier than the higher container in the same group. However, some rehandles may not occur in the actual operation. Therefore, the expected rehandles are proposed to measure the average number of rehandles considering various retrieval sequences of containers in the same group. There would be two expected rehandles for the example of Fig. 3(b). The two types of rehandles are introduced in the next section. In the ICUSP, the stacking distribution changes with the unloading sequence, and different distributions generally result in different numbers of rehandles, as illustrated in Fig. 3(b) and (c). In the distribution of Fig. 3(c), one pessimistic rehandle and one half expected rehandle are produced during the truck retrieving process. In both (b) and (c) of Fig. 3, there are no necessary rehandles since there is no container with lower retrieval priority stacking on the container with higher retrieval priority.

2.2. Mathematical relationship between the yard distribution and the rehandles

Containers with the retrieval priority of m are defined as m-containers. As illustrated in Fig. 4, it is easy to generate various distributions of m-containers in the yard space. Four rehandles are produced pessimistically in both distributions of Fig. 4(a) and (c), and five rehandles are produced pessimistically in the distribution of Fig. 4(b) after the trucks retrieve all m-containers from the yard, defined as the pessimistic rehandles. Pessimistic rehandles are generated in the worst case if the truck retrieving the container at the lower tier arrives before the trucks retrieving the containers stacked on the higher tiers. The pessimistic number of rehandles for retrieving m-containers equals the total number of m-containers in the storage area, excluding the m-containers on the ground tier. It is the same as these containers' retrieval probability by each time; thus, the expected number of rehandles is the mean number of rehandles by retrieving all containers from the yard space, which must be less than the pessimistic number of rehandles. The expected number of rehandles is different among the distributions of Fig. 4(a)–(c).

The two-tier distribution is denoted by (α_1, β_1), as illustrated in Fig. 4(a), or (α_2, β_2), as illustrated in Fig. 4(b), therein showing that there are α_1 or α_2 containers in the first tier and β_1 or β_2 containers in the second tier (the given yard area involves G stacks, $G = \alpha_1 \geq \beta_1 > 0,\ G > \alpha_2 \geq \beta_2 > 0$). Let $P(G, \beta_1)$ and $R(G, \beta_1)$ be the pessimistic and expected number of rehandles when retrieving all containers from the distribution (G, β_1). Similarly, let $P(\alpha_2, \beta_2)$ and $R(\alpha_2, \beta_2)$ be the pessimistic and expected number of rehandles when retrieving all containers from the distribution (α_2, β_2). The three-tier distribution is denoted by $(\alpha_3, \beta_3, \gamma_3)$ in Fig. 4(c), indicating that there are α_3 containers in the first tier, β_3 containers in the second tier, and γ_3 containers in the third tier ($G \geq \alpha_3 \geq \beta_3 \geq \gamma_3$). Let $P(\alpha_3, \beta_3, \gamma_3)$ and $R(\alpha_3, \beta_3, \gamma_3)$ be the pessimistic and expected number of rehandles when retrieving all containers from the distribution $(\alpha_3, \beta_3, \gamma_3)$.

The derivation process for calculating the expected number of rehandles is given in Appendix A, therein taking a three-tier distribution as an example.

Lemma 1. *The slots in the higher tier of the storage space are open for stacking if there are no empty slots in the lower tiers, which is the method used for fast generating the optimal stacking distribution of containers with same retrieval priority.*

Lemma 2. *(i) A container would be rehandled once at most if there are enough yard-stacks for storing rehandled containers. The yard-stacks selected for rehandling the m-containers are not loaded with any m-container initially. When retrieving the m-containers, the number of yard-stacks needed for storing rehandled containers is measured by* $\sum_{\tau=2}^{Num}[(\tau-2)\varphi_m^\tau+1]-[\sum_{\tau=2}^{Num}\varpi(\varphi_m^\tau)-1]$, *depending on the values of* $\varphi_m^2,\ \varphi_m^3,\ \cdots$ *and* φ_m^{Num}. *Here, Num refers to the maximum height (in number of containers) allowed for any yard-stack;* φ_m^τ *refers to the number of yard-stacks including* τ *m-containers (generally,* $\tau = 0, 1, 2, 3, 4$*);* $\varpi(\varphi_m^\tau)$ *equals 1 if* $\varphi_m^\tau > 0$ *and is 0 otherwise. (ii) Based on (i) that there are enough yard-stacks for storing rehandled containers, the expected number of rehandles is* $\sum_{\tau=2}^{Num}\frac{(\tau-1)x_\tau}{\tau}$ *in the yard stacking distribution* $(x_1, x_2, \cdots, x_{Num})$, *in which* x_τ *represents the number of containers in the* τ^{th} *tier of the yard space.*

Table 1
Notations in the IP.

Types	Notations	Meaning
Indices	c	Container (c=1 to C)
	s	Ship-stack (s=1 to S)
	n	Operation (n=1 to N)
	g	Yard-stack (g=1 to G)
	t	Yard-tier (t=1 to Num)
	m	Priority (m=1 to M)
Parameters	C	Number of containers (integer, >0)
	S	Number of ship-stacks (integer, >0)
	N	Number of operations (integer, $=C$)
	G	Number of yard-stacks (integer, >0)
	H_s	Maximum ship-tiers (in number of containers) in ship-stack s (integer, >0)
	M	Number of priority groups (integer, >0)
	Num	Maximum height (in number of containers) allowed for any yard-stack (integer, >1)
	IS_{cs}	=1 if container c is in ship-stack s before the first operation (binary)
	IB_c	Number of containers stacked onto container c (including itself) in the ship before the first operation (integer, >0)
	P_{cm}	=1 if the container c belongs to m-container (binary)
Decision variables	R_{csn}	=1 if container c is removed from the highest available tier of ship-stack s during operation n (binary)
	X_{gmn}	Number of m-containers in yard-stack g before operation n (integer, ≥ 0)
	L_{cgtn}	=1 if container c is loaded into tier t of yard-stack g during operation n (binary)
	B_{cn}	Number of containers stacked onto container c (including itself) in the containership before operation n. It equals 0 if container c has already been taken out of the ship before operation n (integer, ≥ 0)
	C_{cn}	=1 if container c moves (one step) closer to the highest available tier of its ship-stack during operation n (binary)
	E_{csn}	=1 if container c is located in ship-stack s before operation n (binary)
	T_{gtn}	=1 if the slot in tier t of yard-stack g is occupied by a container before operation n (binary)

Lemma 3. *If some containers are distributed into one tier, two tiers or three tiers of the yard space, and the number of containers in the ground tier of the two-tier distribution equals that of the three-tier distribution, then the expected number of rehandles of the three-tier distribution is the highest, and the expected number of rehandles of the one-tier distribution is lower than that of the two-tier distribution.*

The three lemmas show the properties of the proposed ICUSP, and summarize rules of regularity. Lemma 1 proposes a method to generate the optimal stacking distribution and demonstrates the method's effectiveness, which is used to guide the design of the optimization algorithm. Lemma 2 presents an estimation formulation if containers are rehandled once at most. Lemma 3 shows the quantitative relationship of the rehandles and the stacking distributions. The last two lemmas are both the theoretical basis by establishing the objective function in the integer programming model and the internal rationale of the algorithm design. The proofs of the three lemmas are attached in Appendix B.

3. Integer programming model for the ICUSP

3.1. Notations

We now present an integer program (IP) for the ICUSP and use *CPLEX* to solve the model for small-scale instances. Table 1 lists all mathematical notations used in the IP, including the indices, parameters and decision variables.

The ICUSP involves the containership and the terminal yard. The space structure of the containership is composed of the ship-bay, ship-stack and ship-tier. The space structure of the terminal yard is composed of the yard-bay, yard-stack and yard-tier. The index of the ship-tier is removed to reduce the dimensions of the established model. The initial position of container c in the containership is restricted by IS_{cs} and IB_c, and the changing state of container c is tracked by E_{csn} and B_{cn}. E_{csn} and B_{cn} define the located ship-stack and ship-tier of container c before operation n, and they also track the changing state of the ship-bay. Actually, the index of the yard-bay is also removed, which is distinguished by the yard-stacks with continuous integers. However, the index of the yard-tier cannot be removed due to the measurements of the rehandles when the trucks retrieve the containers from the yard. The changing state of the stacking distribution is tracked by L_{cgtn} and T_{gtn}. By reducing the model dimensions, the number of variables is reduced tremendously, which facilitates the fast solving of the mathematical model by the commercial solver.

3.2. Objective function

The objective function of the model can be expressed as

$$\text{minimize} \sum_{g=1}^{G}\sum_{m=1}^{M}(X_{gm(N+1)})^2 \tag{1}$$

Here, $X_{gm(N+1)}$ is a key decision variable used to formulate the stacking distribution, which can be 0, 1, 2, ..., Num in the model. In the four-tier distribution ($Num = 4$), for each g and m, $X_{gm(N+1)}$ can be 0, 1, 2, 3 or 4. To achieve the optimal distribution, the number of $X_{gm(N+1)}$ ($\forall g,\ m$) taking on values of 4, 3 or 2 should be minimized, and the number of $X_{gm(N+1)}$ ($\forall g,\ m$) taking on the value of 1 should be maximized. As demonstrated in Lemma 3, the higher tier distributions of the m-containers ($\forall m$) result in higher expected numbers of rehandles. Therefore, the number of $X_{gm(N+1)}$ taking on a value of 4 should be minimized as the first priority, the number of $X_{gm(N+1)}$ taking on a value of 3 should be minimized as the second priority, and the number of $X_{gm(N+1)}$ taking on a value of 2 should be minimized as the third priority. The priority level of $X_{gm(N+1)}$ is proportional to its actual value. To conclude, the actual value of $X_{gm(N+1)}$ is set as their priority level. Thus, we minimize the summation of $(X_{gm(N+1)})^2$ for all g and m as the objective function, equivalent to the minimization of the expected number of rehandles in the ICUSP.

3.3. Constraints

The constraints are divided into three groups: constraints (2) to (9), constraints (10) to (18), and constraints (19) to (25). The constraints in the first group set limitations on the values of decision variables R_{csn}, r_{sn} and L_{cgtn}.

Group 1:

$$\sum_{n=1}^{N} R_{csn} = IS_{cs}, \forall c, \forall s \tag{2}$$

$$\sum_{c=1}^{C}\sum_{s=1}^{S} R_{csn} = 1, \forall n \tag{3}$$

$$\sum_{c=1}^{C}\sum_{n=1}^{N} R_{csn} \le H_s, \forall s \tag{4}$$

$$\sum_{s=1}^{S}\sum_{n=1}^{N} R_{csn} = 1, \forall c \tag{5}$$

$$\sum_{c=1}^{C}\sum_{g=1}^{G}\sum_{t=1}^{Num} L_{cgtn} = 1, \forall n \tag{6}$$

$$\sum_{g=1}^{G}\sum_{n=1}^{N}\sum_{t=1}^{Num} L_{cgtn} = 1, \forall c \tag{7}$$

$$\sum_{c=1}^{C}\sum_{n=1}^{N}\sum_{t=1}^{Num} L_{cgtn} \le Num, \forall g \tag{8}$$

$$\sum_{s=1}^{S} R_{csn} = \sum_{g=1}^{G}\sum_{t=1}^{Num} L_{cgtn}, \forall c, \forall n \tag{9}$$

Constraints (2) indicate that each container can only be removed from the initial ship-stack. Constraints (3) state that exactly one container is removed from the containership during each operation. Constraints (4) indicate that the number of removing operations in a ship-stack is no greater than the maximum number of tiers in the ship-stack, therein avoiding any shifts in the containership. Constraints (5) show that each container is removed from the containership during an operation. Constraints (6) imply that exactly one container is removed from the containership during each operation. Constraints (7) state that each container is loaded into the yard during the operating process. Constraints (8) ensure that the number of containers loaded into the yard-stack is no greater than the maximum allowed height of the yard-stacks. Constraints (9) ensure that the unloaded container must be loaded into the yard space during each operation.

The constraints in the second group introduce the decision variables B_{cn}, C_{cn} and E_{csn} and connect with decision variables R_{csn}, r_{sn} and L_{cgtn}.

Group 2:

$$B_{c1} = IB_c, \forall c \tag{10}$$

$$B_{c(n+1)} = B_{cn} - C_{cn}, \forall c, \forall n \tag{11}$$

$$\sum_{n=1}^{N} C_{cn} = IB_c, \forall c \tag{12}$$

$$E_{cs1} = IS_{cs}, \forall c, \forall s \tag{13}$$

$$E_{cs(n+1)} = E_{csn} - R_{csn}, \forall c, \forall s, \forall n \tag{14}$$

$$E_{csn} \geq C_{cn} + \sum_{c=1}^{C} R_{csn} - 1, \forall c, \forall s, \forall n \tag{15}$$

$$\sum_{c=1}^{C} R_{csn} \geq E_{csn} + C_{cn} - 1, \forall c, \forall s, \forall n \tag{16}$$

$$C_{cn} \geq \sum_{c=1}^{C} R_{csn} + E_{csn} - 1, \forall c, \forall s, \forall n \tag{17}$$

$$\left(1 - \sum_{s=1}^{S} R_{csn}\right) \cdot (Num - 1) \geq B_{cn} - 1, \forall c, \forall n \tag{18}$$

Constraints (10) initialize B_{cn} with proper values, equivalent to IB_c. Constraints (11) ensure that B_{cn} is updated when containers become closer to the highest available tier of their corresponding ship-stack during each operation. Constraints (12) ensure that the total number of movements closer to the highest available tier of the ship-stack equals IB_c. Constraints (13) initialize E_{csn} with proper values, equivalent to IS_{cs}. Constraints (14) guarantee that E_{csn} is updated when a container is removed from the corresponding ship-stack for each operation. Constraints (15) to (17) define the relationships among E_{csn}, r_{sn} and C_{cn}: if any two of E_{csn}, r_{sn} and C_{cn} equal 1, the third also equals 1. Constraints (18) indicate that a container can be removed from a ship-stack only if the container is located on the highest available tier of the ship-stack.

The constraints in the last group attempt to reduce the number of rehandles when stacking containers into the same yard-stack.

Group 3:

$$\sum_{n=1}^{N}\sum_{c=1}^{C} n \cdot L_{cgtn} + 1 \leq \sum_{n=1}^{N} (n+1) \cdot (1 - T_{g(t+1)(n+1)}) + \sum_{n=1}^{N}\sum_{c=1}^{C} n \cdot L_{cg(t+1)n}, \forall g, \forall t \in [1, Num - 1] \tag{19}$$

$$T_{gtn} \geq T_{g(t+1)n}, \forall g, \forall t \in [1, Num - 1], \forall n \in [1, N + 1] \tag{20}$$

$$\sum_{n=1}^{N}\sum_{c=1}^{C}\sum_{m=1}^{M} m \cdot P_{cm} \cdot L_{cgtn} \geq \sum_{n=1}^{N}\sum_{c=1}^{C}\sum_{m=1}^{M} m \cdot P_{cm} \cdot L_{cg(t+1)n}, \forall g, \forall t \in [1, Num - 1] \tag{21}$$

$$T_{gt1} = 0, \forall g, \forall t \tag{22}$$

$$T_{gt(n+1)} = T_{gtn} + \sum_{c=1}^{C} L_{cgtn}, \forall g, \forall t, \forall n \tag{23}$$

$$X_{gm1} = 0, \forall g, \forall m \tag{24}$$

$$X_{gm(n+1)} = X_{gmn} + \sum_{c=1}^{C}\sum_{t=1}^{Num} P_{cm} \cdot L_{cgtn}, \forall g, \forall m, \forall n \tag{25}$$

Constraints (19) and (20) ensure the slot in the lower yard-tier is loaded first by obtaining the values of L_{cgtn}, $L_{cg(t+1)n}$, T_{gtn} and $T_{g(t+1)n}$. Constraints (21) ensure that the priority of a container in the higher yard-tier is not greater than the

Table 2
Computational results for the two methods on small-scale instances.

Instances scale/*C*	*G*	*Num*	*S*	*CPLEX*		*TS*	
				objective	$t(s)$	objective	$t(s)$
6	2	4	3	9.80	4.28	9.80	<0.01
7	3	4	3	0.82	15.82	0.82	<0.01
8	3	4	3	9.90	47.10	9.90	<0.01
9	3	4	3	14.00	120.31	14.00	<0.01
10	4	4	3	11.60	601.80	11.60	<0.01
11	4	4	3	14.00	3204.39	14.00	<0.01
12	4	4	3	18.86	1707.90	18.86	0.01

priority of a container in the lower yard-tier. The constraint is always satisfied if $S \geq Num$, which can be ensured by *the initial phase* in the next section. Constraints (22) initialize T_{gtn} to 0 because no container is located in the yard-stacks before the first operation. Constraints (23) ensure that T_{gtn} is updated when any container is loaded into the slot in tier t of yard-stack g for each operation. Constraints (24) initialize X_{gmn} to 0 when the initial yard space is empty. Constraints (25) ensure that X_{gmn} is updated when any m-container is loaded into yard-stack g at operation n.

3.4. Computational results by solving the IP using CPLEX

The mathematical model for the ICUSP is solved by *IBM ILOG CPLEX Optimization Studio* 12.6. A total of 70 instances are considered - 10 each scaled with 6, 7, 8, 9, 10, 11 and 12 containers. All the results in this study are obtained using a computer with 4 gigabytes of RAM, the Windows 7 Professional 32-bit operating system, and an Intel Core i7 processor with two 2.8 GHz cores. The first four columns show the values of the input parameters for each scales of instances. The 5th and 6th columns list the average value of the objective and the average runtime needed by *CPLEX* of 10 instances. The last two columns show the average value of the objective and the average runtime of 10 instances, using the two-stage search algorithm (*TS*) programmed solved by Matlab (introduced in the next section). The last line of Table 2 shows the average value of seven instances because three instances cannot be solved by *CPLEX* within a limited time period. The results for each instance are attached in Appendix C for reference.

The results indicate that the mathematical model can be solved by *CPLEX* for most of small-scale instances, and the two-stage search algorithm can find the optimal solution quickly less than 0.02s for the given small-scale instances. Referring to Appendix C, *CPLEX* needs more than an hour to find an optimal solution for instances 11-8, 11-9, 11-10 and 12-7. Moreover, it cannot obtain results for instances 12-8, 12-9 and 12-10 within 5 h. Thus, an efficient algorithm is needed for solving large-scale instances. The limitation of *CPLEX* on solving efficiency has significant effects on the practices.

4. Two-stage search algorithm

To achieve the high solving efficiency for the large-scale instances, this study proposes a two-stage search algorithm (*TS*) to solve the ICUSP.

4.1. The initial stage

The initial stage focuses on a feasible solution to the ICUSP, avoiding the necessary rehandles. The basic idea can be outlined as follows: Choose several of the highest ship-stacks, unload a container from each of the selected highest ship-stacks, and stack them to a yard-stack in descending order of retrieval priority. Update the heights of the ship-stacks, and continue to choose several of the highest ship-stacks until all the ship-stacks are empty.

Fig. 5 illustrates the process of *the initial stage*, which is further explained using an instance in Appendix D. Note that $|A|$ represents the number of elements in set A.

4.2. The neighborhood search stage

The neighborhood search stage is used to optimize the feasible solution by *the initial stage*. In the feasible solution, some m-containers may be stacked in the same yard-stack. This stage attempts to exchange one of the m-containers with a container in a neighboring yard-stack, ensuring that the m-containers are distributed in different yard-stacks to the greatest extent. *The neighborhood search stage* focuses on reducing the expected number of rehandles when the trucks retrieve containers from the yard.

Fig. 6 illustrates the process of *the neighborhood search stage*.

Evaluation rule. If the pessimistic number of rehandles does not increase when retrieving containers from the changed yard-stacks and no rehandling occurs in the container ship when unloading containers, then accept the new distribution and update the stacking distribution by $Yard_0$.

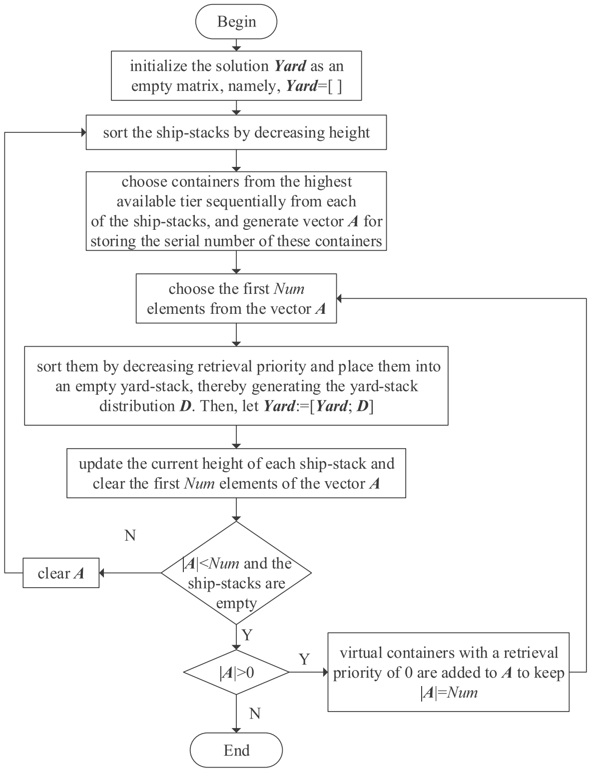

Fig. 5. Flow chart for *the initial stage*.

The ***Evaluation rule*** avoids rehandling in the containership when unloading containers or decreasing the number of rehandles in the yard when retrieving containers. Taking Fig. 3(b) as an example, there are two 4-containers in the first yard-stack. The containers in the second yard-stack are from different groups compared with those in the first yard-stack. If exchanging {4: [4, 5], 1} with {2: [4, 4], 2} and reordering the containers in each of the two yard-stacks, the unloading sequence of the first eight containers would be {7: [3, 5], 1}, {4: [5, 5], 1}, {3: [2, 5], 1}, {2: [4, 4], 2}, {6: [6, 5], 2}, {5: [1, 5], 2}, {4: [4, 5], 1}, {1: [3, 4], 2}, where the rehandle is produced in the fourth ship-stack because the 2-container in the fourth tier would be unloaded before the 4-container in the fifth tier. The exchange cannot satisfy the ***Evaluation rule***, and thus, we reject the new solution. If exchanging {4: [5, 5], 1} with {1: [3, 4], 2} and sorting the containers in each of the two yard-stacks, the unloading sequence of the first eight containers would be {7: [3, 5], 1}, {4: [4, 5], 1}, {3: [2, 5], 1}, {1: [3, 4], 2}, {6: [6, 5], 2}, {5: [1, 5], 2}, {4: [5, 5], 1}, {2: [4, 4], 2}, which satisfy the ***Evaluation rule***. Therefore, we accept the new solution. The ***Evaluation rule*** ensures that the solution is strictly constrained by the containership stowage plan.

Theorem 1. *The proposed TS for the ICUSP is convergent.*

Proof of Theorem 1. *The initial stage*, the 1st stage of the *TS*, can obtain a feasible solution without necessary rehandles based on **Property**. *The neighborhood search stage*, the 2nd stage of the *TS*, is designed to optimize the stacking distribution to minimize the expected number of rehandles. The stage assigns the containers with same retrieval priority to different yard-stacks by iteration and search techniques. The stacking distribution would be from the three-tier distribution to the two-tier distribution, and from the two-tier distribution to the one-tier distribution for each groups of containers. As proved by Lemma 3, $R(\alpha_2, \beta_2, \gamma_2) > R(\alpha_1, \beta_1) > R(\alpha_0)$. The stacking distribution is optimized from $(\alpha_2, \beta_2, \gamma_2)$ to (α_1, β_1), and from (α_1, β_1) to (α_0).

The *TS* optimizes the distribution by moving a m-container from a yard-stack including α m-containers to a yard-stack including β m-containers ($\alpha >= 2$, $\beta >= 0$, and $\alpha - \beta > 1$). According to Lemma 2 (ii), the gap of expected rehandles by movement is denoted by $\Delta = -\frac{\alpha-1}{\alpha} + \frac{\beta}{\beta+1} = \frac{1-(\alpha-\beta)}{\alpha(\beta+1)} < 0$. Thus, the method achieves the convergence. □

Theorem 2. *By applying the two-stage search algorithm, the optimal yard distribution can be found in polynomial time, equivalent to $O(N^2)$.*

Proof of Theorem 2. To present the time complexity of the two-stage search algorithm, we first calculate the time complexity of each stage. The computational complexity of the first stage is measured by the product of the number of iterations $\frac{G}{4}$ and the number of the complexity of the sorting problem $S^2 + Num^2$. The computational complexity of the second stage

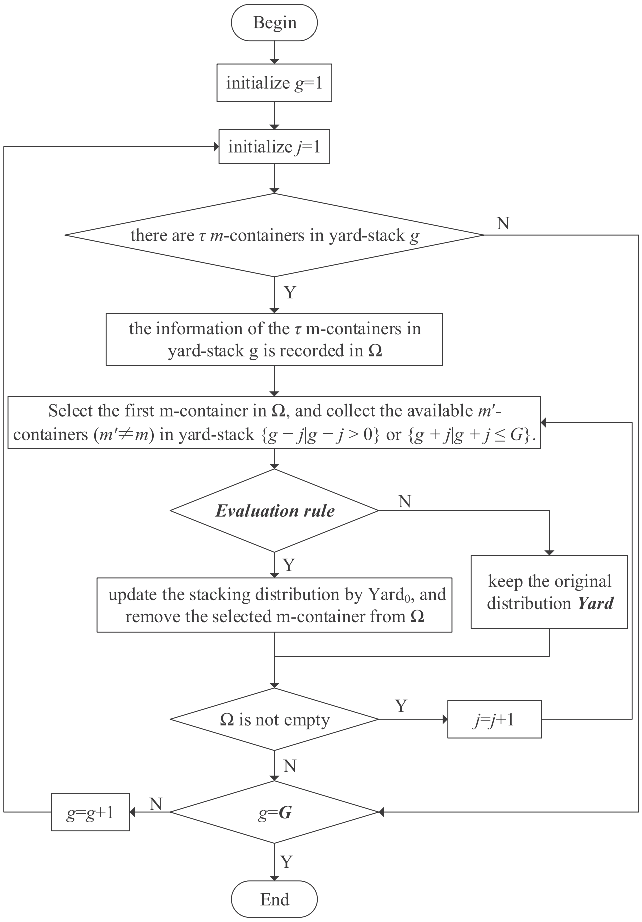

Fig. 6. Flow chart for *the neighborhood search stage*.

summarizes the time complexity for each group of containers' unloading and stacking operations, measured by the total number of iterations for optimizing the distribution of all groups of containers. The time complexity for distributing each group of containers is measured by the product of the maximum number of steps in searching an exchanged container for each rehandled container and the number of containers that need to be rehandled. For each rehandled container, $G-1$ is the maximum number of steps for the **Evaluation rule**, which refers to the number of available yard-stacks for exchanging. In the stacking distribution, there are $2\varphi_m^3+\varphi_m^2$ containers that need to be exchanged with other containers. Because $\varphi_m^3+\varphi_m^2+\varphi_m^1+\varphi_m^0 = G$, $3\varphi_m^3+2\varphi_m^2+\varphi_m^1 = d_m$ (d_m represents the number of m-containers), and $\sum_{m=1}^{M}\varphi_m^0=N$, the upper bound of the number of iterations for each group of containers equals $(G-1)\cdot(d_m-G+\varphi_m^0)$). In addition, the time complexity of the second stage is the summation of the number of iterations for all container groups, that is, $\sum_{m=1}^{M}(G-1)\cdot(d_m-G+\varphi_m^0)$). The time complexity of the algorithm is $O(\frac{G}{4}(S^2+Num^2)+(G-1)*(N-M*G+\sum_{m=1}^{M}\varphi_m^0))$. Because $O(G)=O(N)$, the conclusion can be drawn that $O(\frac{G}{4}(S^2+Num^2)+(G-1)*(N-M*G+\sum_{m=1}^{M}\varphi_m^0))$ $=O(\frac{G}{4}(S^2+Num^2)+(G-1)*N) = O(G*N)$ $= O(N^2)$. The theorem is thus proved. □

5. Computational experiments

To test the efficiency of the *TS*, we compare the results obtained by the *TS* and *CPLEX* in Section 3. The results in Table 2 show that the objective value given by the *TS* is the same as that given by *CPLEX* for each small-scale instance; therefore, the *TS* is sufficiently efficient to achieve an optimal solution for small-scale instances. However, the runtime of the *TS* for each instance nearly approaches zero, making it much less than the runtime of *CPLEX*. With increasing instance scale, the *TS* is superior to *CPLEX* in terms of runtime.

Another numerical experiment is conducted to verify the efficiency of the *TS*, therein involving four types of yard sizes and six types of ship-bay sizes by considering 8 time windows. The step length ranges from 1 to 10, and the results are presented in Table 3. The ship-bay size is defined by the height of the ship-stack (H) and the number of ship-stacks (S), and each ship-bay size corresponds to four different types of yard sizes defined by the height of the yard-stacks (Num) and the

Table 3
Expected number of rehandles required on four types of yard sizes and six of ship-bay sizes (8 time windows).

Ship-bay size	Yard size	Average expected number of rehandles for 100 instances									
$H \times S$	$Num \times G$	AT(s)	NS-0	NS-1	NS-2	NS-3	NS-4	NS-5	NS-6	...	NS-10
10 × 5	2 × 25	0.02	1.59	0.13	**0.04**	0.04	0.04	0.04	0.04	0.04	0.04
	3 × 17	0.04	2.78	0.78	**0.61**	0.61	0.61	0.61	0.61	0.61	0.61
	4 × 13	0.09	4.32	2.12	**2.09**	2.09	2.09	2.09	2.09	2.09	2.09
	5 × 10	0.10	5.77	**4.23**	4.23	4.23	4.23	4.23	4.23	4.23	4.23
20 × 5	2 × 50	0.04	3.10	0.17	**0.03**	0.03	0.03	0.03	0.03	0.13	0.03
	3 × 34	0.11	6.07	1.42	**1.21**	1.21	1.21	1.21	1.21	1.21	1.21
	4 × 25	0.26	8.89	4.81	**4.76**	4.76	4.76	4.76	4.76	4.76	4.76
	5 × 20	0.37	11.82	**8.52**	8.52	8.52	8.52	8.52	8.52	8.52	8.52
10 × 10	2 × 50	0.04	3.11	0.17	0.02	**0**	0	0	0	0	0
	3 × 34	0.05	6.05	0.95	0.30	0.20	**0.19**	0.19	0.19	0.19	0.19
	4 × 25	0.10	8.97	2.56	1.39	**1.29**	1.29	1.29	1.29	1.29	1.29
	5 × 20	0.20	11.47	4.93	3.92	**3.91**	3.91	3.91	3.91	3.91	3.91
20 × 10	2 × 100	0.11	6.32	0.34	0.03	**0.01**	0.01	0.01	0.01	0.01	0.01
	3 × 67	0.12	12.09	1.63	0.38	0.19	**0.17**	0.17	0.17	0.17	0.17
	4 × 50	0.25	17.68	5.14	2.90	2.66	**2.65**	2.65	2.65	2.65	2.65
	5 × 40	0.52	23.13	10.16	8.10	**8.07**	8.07	8.07	8.07	8.07	8.07
10 × 15	2 × 75	0.07	4.76	0.38	0.03	0.01	**0**	0	0	0	0
	3 × 50	0.07	8.82	1.39	0.33	0.11	0.07	**0.05**	0.05	0.05	0.05
	4 × 38	0.10	13.13	3.80	1.45	0.85	**0.76**	0.76	0.76	0.76	0.76
	5 × 30	0.22	17.21	7.19	3.97	3.34	**3.30**	3.30	3.30	3.30	3.30
20 × 15	2 × 150	0.22	9.11	0.61	0.04	0.01	0.01	**0**	0	0	0
	3 × 100	0.21	18.49	3.00	0.71	0.22	0.13	**0.10**	0.10	0.10	0.10
	4 × 75	0.33	26.28	7.41	2.93	1.74	1.54	**1.53**	1.53	1.53	1.53
	5 × 60	0.63	34.87	14.71	8.04	6.69	6.61	**6.60**	6.60	6.60	6.60

number of yard-stacks (G). A total of 100 different instances are generated for each type of ship-bay size, and each value from the third to the last column is the average value of 100 instances. AT represents the average runtime for 100 instances in 10 steps of the neighborhood search. NS-j represents the results by j steps, in which NS-0 represents the initial results by *the initial phase*. The column for NS-10 shows the expected number of rehandles by 10 steps.

The *TS* can find the optimal solution of all instances within 10 steps in less than 1 second. The expected number of rehandles becomes stable after 5 steps. The decreasing gap usually appears in NS-1, NS-2 and NS-3, and tiny improvements are achieved in the later steps. The results show that the convergence step j varies seemingly depending on the S. When S=5, the objective converges to a stable value in 1st or 2nd step. When S=10, the objective converges to a stable value in 3th or 4th step. When S=15, the objective converges to a stable value in 4th or 5th step. The minimum expected number of rehandles depends on H and increases with increasing yard-stack height, which coincides with the estimation formulation in Lemma 2.

The arrival time windows length of the trucks is an important parameter to be further explored. A large set of instances are generated randomly for different scenarios of time windows to see how the expected number of rehandles and convergence step behave in various time window length. Table 4 records the average expected number of rehandles by each neighbourhood search step for ship-bay size 20 × 15 in 4, 6, 8, 10, and 12 time windows, respectively. In which, the experimental results of 8 time windows are extracted form Table 3. For the given set of trucks, the less the number of time windows, the longer the time window length. The experimental results indicate that the time window length has an influence on the expected number of rehandles and convergence step. With the decrease of time window length, the expected number of rehandles and convergence step both decrease. The results give some insights on the importance of a good arrival forecast. All experimental results about the sensitivity analysis of the time window length for all ship-bay sizes in Table 3 can be found in Appendix E.

In the numerical experiment, the instances scale reaches 300 containers, but the runtime is less than 1 second. Thus, the *TS* is efficient in solving large-scale instances.

To demonstrate the complexity of the *TS* by numerical instances, 1000 instances are generated randomly for each scale, and the range of the instance scale is from 15 to 450 containers, with a scale interval of 15. Fig. 7 illustrates the relationship between the runtime (y) and the scale (x). The fitting function is $y = 0.006x^2 - 0.5421x + 31.316$, fitted as a polynomial function. This shows that the runtime increases polynomially with increasing scale. This obviously coincides with the time complexity in Theorem 2.

In Fig. 7, the largest instances scale is up to 480 container, which can be solved in approximately 1 second. Thus, the validity of the algorithm is demonstrated again. The current largest containership has a capacity of approximately 21000 TEUs and has a length of 399.9 m, a width of 58.6 m, and a depth of 33.5 m. It can be inferred that there are approximately 60 ship-bays, and each ship-bay has a capacity of approximately 400 TEU. Thus, the *TS* is efficient for real-world applications.

Table 4
Sensitivity analysis of time window length on the expected number of rehandles with ship-bay size 20 × 15.

Time windows	Yard size	Average expected number of rehandles for 100 instances									
	Num × *G*	AT(s)	NS-0	NS-1	NS-2	NS-3	NS-4	NS-5	NS-6	NS-7	...
4	2 × 150	0.12	18.83	3.73	1.18	0.47	0.16	0.07	0.04	**0.02**	0.02
	3 × 100	0.54	35.80	15.71	9.53	6.66	5.01	4.61	**4.59**	4.59	4.59
	4 × 75	0.76	50.67	32.11	25.37	22.41	21.78	21.73	**21.72**	21.72	21.72
	5 × 60	0.71	62.85	52.42	48.17	**47.32**	47.32	47.32	47.32	47.32	47.32
6	2 × 150	0.08	12.49	1.35	0.23	0.02	0.01	**0**	0	0	0
	3 × 100	0.11	23.72	5.59	2.05	0.85	0.40	**0.31**	0.31	0.31	0.31
	4 × 75	0.37	34.62	13.98	7.42	4.94	4.58	4.55	**4.54**	4.54	4.54
	5 × 60	0.55	44.85	24.36	17.25	15.77	15.70	**15.69**	15.69	15.69	15.69
8	2 × 150	0.22	9.11	0.61	0.04	0.01	0.01	**0**	0	0	0
	3 × 100	0.21	18.49	3.00	0.71	0.22	0.13	**0.10**	0.10	0.10	0.10
	4 × 75	0.33	26.28	7.41	2.93	1.74	1.54	**1.53**	1.53	1.53	1.53
	5 × 60	0.63	34.87	14.71	8.04	6.69	6.61	**6.60**	6.60	6.60	6.60
10	2 × 150	0.08	7.23	0.27	0.02	**0**	0	0	0	0	0
	3 × 100	0.07	14.71	1.62	0.28	0.08	0.04	**0.03**	0.03	0.03	0.03
	4 × 75	0.13	21.55	4.48	1.42	0.67	0.59	**0.58**	0.58	0.58	0.58
	5 × 60	0.34	27.72	9.18	4.13	3.20	3.16	**3.15**	3.15	3.15	3.15
12	2 × 150	0.10	6.12	0.22	**0**	0	0	0	0	0	0
	3 × 100	0.07	12.55	0.96	0.16	0.05	**0.02**	0.02	0.02	0.02	0.02
	4 × 75	0.10	17.89	3.04	0.65	0.28	**0.23**	0.23	0.23	0.23	0.23
	5 × 60	0.26	23.65	6.35	2.22	1.67	1.64	**1.63**	1.63	1.63	1.63

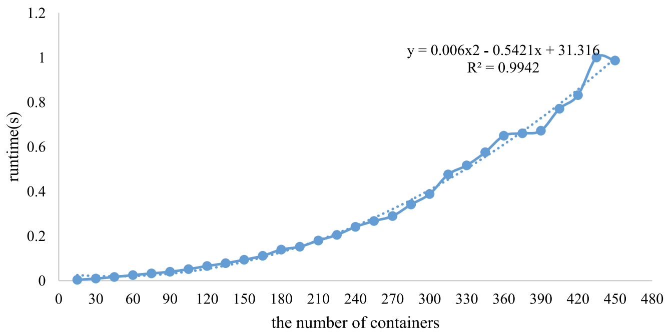

Fig. 7. The increasing trend of the runtime with increasing instance scale.

6. Conclusions

The ICUSP is a new proposed optimization problem, which can be applied to the development of automated container terminals by reducing the rehandles from the source. An integer model and two-stage search algorithm was developed to obtain the reasonable unloading sequence and optimal yard stacking distribution simultaneously. The model can only be solved by CPLEX for small-scale instances, so we designed the *TS* to meet practical requirements. *The initial stage* in the *TS* avoids the necessary rehandles considering **Property** of the ICUSP, while *the neighborhood search stage* in the *TS* aims to minimize the expected number of rehandles using internal mechanism in Lemma 3. The quantitative relationship between the expected number of rehandles and the stacking height was given in the estimation formulation in Lemma 2, which ensures the convergence of the *TS*. The experimental results showed that the *TS* could achieve optimal solutions for a wide variety of randomly generated numerical instances. The arrival time windows length of the trucks is an important factor on the expected number of rehandles. In particular, the *TS* could find optimal solutions for large-scale instances in polynomial time. The computation time of the two-stage search algorithm was less than 1 s for instances of up to 300 containers. The numerical experiment of a large number of instances demonstrated the time complexity of the proposed algorithm, which coincides with the theoretical results.

Future works would focus on the integration of quay crane and yard crane based on the stacking distribution, making the operation plan of these handling equipment [33,34]. We can also study the influence of double cycling of yard crane in the yard on the stacking distribution.

Funding

This work was supported by the National Natural Science Foundation of China, 71971035 and 71572022; Liaoning BaiQianWan Talents Program, 2016236; and The Fundamental Research Funds for the Central Universities, 3132019021.

Appendix A. The derivation process

The derivation process is a full enumeration by retrieving containers one by one from the storage yard. To implement the estimation of the expected number of rehandles in a stacking distribution intuitively, we take a three-tier distribution as an example and present the derivation process here.

The three-tier distribution is illustrated as distribution (3,2,1) in Fig. A.1. Three containers, including containers 1, 2, and 3, are located in the first tier; two containers, including containers 4 and 5, are located in the second tier; and only one container, container 6, is located in the third tier. The yard space consists of six yard-stacks, and six containers are distributed across only three of them. The other distributions in Fig. A.1 are the resulting distributions in the retrieving process, where the containers are renumbered after each retrieval. The expected number of rehandles for the three-tier distribution is denoted by $R(3, 2, 1)$ for $G = 6$.

After retrieving one container randomly from the distribution (3,2,1), the resulting distributions could be (3,2,0), (3,1,1), (2,2,1) or (4,1,0). The probability of each container being retrieved is the same: one in six. If retrieving container 6 from the distribution (3,2,1), no rehandle occurs, and the resulting distribution would be (3,2,0). If retrieving container 5 from the distribution (3,2,1), no rehandle occurs, and the resulting distribution would be (3,1,1). If retrieving container 2 from the distribution (3,2,1), one rehandle occurs, and the resulting distribution would also be (3,1,1). The resulting distributions are all listed in the first column of Table A.1 after retrieving one container from the stacking distribution (3,2,1). The second column represents the probability of generating the corresponding distribution. The last column records the retrieved container number and the number of rehandles when retrieving this container.

Thus, the expected number of rehandles for the distribution (3,2,1) can be computed considering the resulting distributions. The iterative formulation is listed here.

$$\begin{aligned} R(3,2,1) &= \frac{1}{6}[0 + R(3,2,0)] + \frac{1}{6}[0 + R(3,1,1)] + \frac{1}{6}[1 + R(3,1,1)] + \frac{1}{6}[0 + R(2,2,1)] \\ &+ \frac{1}{6}[1 + R(4,1,0)] + \frac{1}{6}[2 + R(4,1,0)] \\ &= \frac{2}{3} + \frac{1}{6}R(3,2,0) + \frac{1}{3}R(3,1,1) + \frac{1}{6}R(2,2,1) + \frac{1}{3}R(4,1,0) \end{aligned}$$

In a similar manner, the iteration continues until the expected number of rehandles is obtained. Tables A.1–A.11 present the complete derivation process.

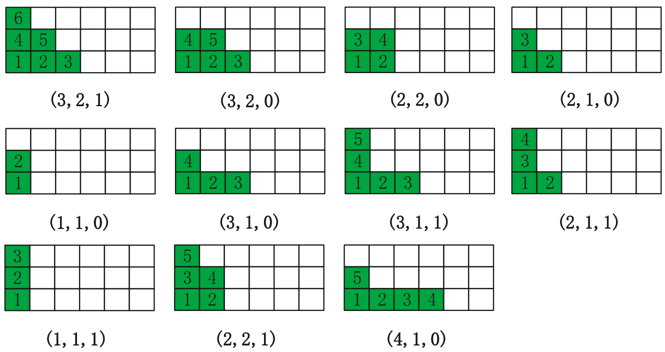

Fig. A.1. The original and intermediate stacking distributions in the retrieving process (the containers are renumbered in each stacking distribution).

Table A.1
Resulting distributions after one retrieval from (3,2,1).

Distribution	Probability	Retrieved container No.# (number of rehandles)
(3,2,0)	1/6	6(0)
(3,1,1)	1/3	5(0), 2(1)
(2,2,1)	1/6	3(0)
(4,1,0)	1/3	4(1), 1(2)

Table A.2
Resulting distributions after one retrieval from (3,2,0).

Distribution	Probability	Retrieved container No.# (number of rehandles)
(2,2,0)	1/5	3(0)
(3,1,0)	4/5	4(0), 5(0), 1(1), 2(1)

Table A.3
Resulting distributions after one retrieval from (2,2,0).

Distribution	Probability	Retrieved container No.# (number of rehandles)
(2,1,0)	1	3(0), 4(0), 1(1), 2(1)

Table A.4
Resulting distributions after one retrieval from (2,1,0).

Distribution	Probability	Retrieved container No.# (number of rehandles)
(1,1,0)	1/3	2(0)
(2,0,0)	2/3	3(0), 1(1)

Table A.5
Resulting distributions after one retrieval from (1,1,0).

Distribution	Probability	Retrieved container No.# (number of rehandles)
(1,0,0)	1	2(0), 1(1)

Table A.6
Resulting distributions after one retrieval from (3,1,0).

Distribution	Probability	Retrieved container No.# (number of rehandles)
(3,0,0)	2/4	4(0), 1(1)
(2,1,0)	2/4	2(0), 3(0)

Table A.7
Resulting distributions after one retrieval from (3,1,1).

Distribution	Probability	Retrieved container No.# (number of rehandles)
(3,1,0)	1/5	5(0)
(2,1,1)	2/5	2(0), 3(0)
(4,0,0)	2/5	4(1), 1(2)

Table A.8
Resulting distributions after one retrieval from (2,1,1).

Distribution	Probability	Retrieved container No.# (number of rehandles)
(2,1,0)	1/4	4(0)
(1,1,1)	1/4	2(0)
(3,0,0)	2/4	3(1), 1(2)

Table A.9
Resulting distributions after one retrieval from (1,1,1).

Distribution	Probability	Retrieved container No.# (number of rehandles)
(1,1,0)	1/3	3(0)
(2,0,0)	2/3	2(1), 1(2)

Table A.10
Resulting distributions after one retrieval from (2,2,1).

Distribution	Probability	Retrieved container No.# (number of rehandles)
(2,2,0)	1/5	5(0)
(2,1,1)	2/5	4(0), 2(1)
(3,1,0)	2/5	3(1), 1(2)

Table A.11
Resulting distributions after one retrieval from (4,1,0).

Distribution	Probability	Retrieved container No.# (number of rehandles)
(4,0,0)	2/5	5(0), 1(1)
(3,1,0)	3/5	2(0), 3(0), 4(0)

$$\begin{aligned}
R(3,2,0) &= \frac{1}{5}[0+R(2,2,0)]+\frac{2}{5}[0+R(3,1,0)]+\frac{2}{5}[1+R(3,1,0)]\\
&= \frac{2}{5}+\frac{1}{5}R(2,2,0)+\frac{4}{5}R(3,1,0)\\
R(2,2,0) &= \frac{2}{4}[0+R(2,1,0)]+\frac{2}{4}[1+R(2,1,0)]\\
&= \frac{1}{2}+R(2,1,0)\\
R(2,1,0) &= \frac{1}{3}[0+R(1,1,0)]+\frac{1}{3}[0+R(2,2,0)]+\frac{1}{3}[1+R(2,0,0)]\\
&= \frac{1}{3}+\frac{1}{3}R(1,1,0)+\frac{2}{3}R(2,0,0)\\
&= \frac{1}{3}+\frac{1}{3}R(1,1,0)\\
R(1,1,0) &= \frac{1}{2}[0+R(1,0,0)]+\frac{1}{2}[1+R(1,0,0)]\\
&= \frac{1}{2}+R(1,0,0)\\
&= \frac{1}{2}\\
R(3,1,0) &= \frac{1}{4}[0+R(3,0,0)]+\frac{1}{4}[1+R(3,0,0)]+\frac{2}{4}[0+R(2,1,0)]\\
&= \frac{1}{4}+\frac{1}{2}R(3,0,0)+\frac{1}{2}R(2,1,0)\\
&= \frac{1}{4}+\frac{1}{2}R(2,1,0)\\
R(3,1,1) &= \frac{1}{5}[0+R(3,1,0)]+\frac{2}{5}[0+R(2,1,1)]+\frac{1}{5}[1+R(4,0,0)]+\frac{1}{5}[2+R(4,0,0)]\\
&= \frac{3}{5}+\frac{1}{5}R(3,1,0)+\frac{2}{5}R(2,1,1)+\frac{2}{5}R(4,0,0)\\
&= \frac{3}{5}+\frac{1}{5}R(3,1,0)+\frac{2}{5}R(2,1,1)\\
R(2,1,1) &= \frac{1}{4}[0+R(2,1,0)]+\frac{1}{4}[0+R(1,1,1)]+\frac{1}{4}[1+R(3,0,0)]+\frac{1}{4}[2+R(3,0,0)]\\
&= \frac{3}{4}+\frac{1}{4}R(2,1,0)+\frac{1}{4}R(1,1,1)+\frac{2}{4}R(3,0,0)\\
&= \frac{3}{4}+\frac{1}{4}R(2,1,0)+\frac{1}{4}R(1,1,1)\\
R(1,1,1) &= \frac{1}{3}[0+R(1,1,0)]+\frac{1}{3}[1+R(2,0,0)]+\frac{1}{3}[2+R(2,0,0)]\\
&= 1+\frac{1}{3}R(1,1,0)+\frac{2}{3}R(2,0,0)\\
&= 1+\frac{1}{3}R(1,1,0)\\
R(2,2,1) &= \frac{1}{5}[0+R(2,2,0)]+\frac{1}{5}[0+R(2,1,1)]+\frac{1}{5}[1+R(2,1,1)]+\frac{1}{5}[1+R(3,1,0)]+\frac{1}{5}[2+R(3,1,0)]\\
&= \frac{4}{5}+\frac{1}{5}R(2,2,0)+\frac{2}{5}R(2,1,1)+\frac{2}{5}R(3,1,0)\\
R(4,1,0) &= \frac{1}{5}[0+R(4,0,0)]+\frac{1}{5}[1+R(4,0,0)]+\frac{3}{5}[0+R(3,1,0)]\\
&= \frac{1}{5}+\frac{2}{5}R(4,0,0)+\frac{3}{5}R(3,1,0)\\
&= \frac{1}{5}+\frac{3}{5}R(3,1,0)
\end{aligned}$$

Finally, we can obtain the conclusion that $R(3,2,1)=\frac{5}{3}$. According to Lemma 2 (ii), R(3, 2, 1) should be the summation of $2\times\frac{1}{2}$ and $1\times\frac{2}{3}$, and the result obtained by the formulation is $\frac{5}{3}$, identical to the derived results.

Appendix B. Proof of Lemmas

Lemma 1. *The slots in the higher tier of the storage space are open for stacking if there are no empty slots in the lower tiers, which is the method used for fast generating the optimal stacking distribution of containers with same retrieval priority.*

Proof of Lemma 1. The distributions in Fig. 4 are used as examples, in which the distribution of Fig. 4(a) is generated according to the above method in Lemma 1, and the distributions of Fig. 4(b) and (c) are generated randomly.

(i) Clearly, $P(\alpha_1,\ \beta_1) = P(\alpha_3,\ \beta_3,\ \gamma_3)\ <\ P(\alpha_2,\ \beta_2)$.

We can draw the conclusion that the distribution of Fig. 4(b) is the worst and has the maximum pessimistic number of rehandles.

(ii) As proven in Yu and Qi (2013), $R(G, \beta_1) = \frac{\beta_1(G+2)}{2(G+1)}$ for $\alpha_1 = G$, as illustrated in Fig. 4(a), and $R(\alpha_2, \beta_2) = \frac{\beta_2}{2}$, as illustrated in Fig. 4(b). Assuming that $\alpha_2 + \beta_2 = \alpha_1 + \beta_1,\ \beta_2 \le \alpha_2$.

If $\alpha_2 < \alpha_1 = G$ and $\beta_1 < \beta_2$, then $R(G, \beta_1) - R(\alpha_2, \beta_2) = \frac{\beta_1(G+2)}{2(G+1)} - \frac{\beta_2}{2} = \frac{(\alpha_2-\alpha_1)G+2(\alpha_2-\alpha_1)+\beta_2}{2(G+1)}$. If $\alpha_2 < \alpha_1$, then $2(\alpha_2 - \alpha_1) \le -2$. If $G > 0$, then $G(\alpha_2 - \alpha_1) \le -G$. Moreover, if $\beta_2 \le \alpha_2 < \alpha_1$, then $(\alpha_2 - \alpha_1)G + 2(\alpha_2 - \alpha_1) + \beta_2 < -2$. Thus, $R(G, \beta_1) - R(\alpha_2, \beta_2) < 0$.

Briefly, $R(\alpha_1, \beta_1) - R(\alpha_2, \beta_2) < 0$, namely, $R(\alpha_1,\ \beta_1) < R(\alpha_2,\ \beta_2)$. The distribution with the fewest expected rehandles is generated when containers are distributed to the largest number of yard-stacks.

A comparison of Fig. 4(a) and (c) is provided in Lemma 3, and the result is that $R(\alpha_2,\ \beta_2) < R(\alpha_3,\ \beta_3,\ \gamma_3)$.

In conclusion, $R(\alpha_1,\ \beta_1) \le R(\alpha_2,\ \beta_2) < R(\alpha_3,\ \beta_3,\ \gamma_3)$, that is, the distribution of Fig. 4(a) is the best and the distribution of Fig. 4(b) is the worst based on the expected number of rehandles. □

Lemma 2. *(i) A container would be rehandled once at most if there are enough yard-stacks for storing rehandled containers. The yard-stacks selected for rehandling the m-containers are not loaded with any m-container initially. When retrieving the m-containers, the number of yard-stacks needed for storing rehandled containers is measured by* $\sum_{\tau=2}^{Num}[(\tau-2)\varphi_m^\tau + 1] - [\sum_{\tau=2}^{Num}\varpi(\varphi_m^\tau) - 1]$*, depending on the values of* $\varphi_m^2,\ \varphi_m^3,\ \cdots$ *and* φ_m^{Num}*. Here, Num refers to the maximum height (in number of containers) allowed for any yard-stack;* φ_m^τ *refers to the number of yard-stacks including* τ *m-containers (generally,* $\tau = 0, 1, 2, 3, 4$*);* $\varpi(\varphi_m^\tau)$ *equals 1 if* $\varphi_m^\tau > 0$ *and is 0 otherwise. (ii) Based on (i) that there are enough yard-stacks for storing rehandled containers, the expected number of rehandles is* $\sum_{\tau=2}^{Num}\frac{(\tau-1)x_\tau}{\tau}$ *in the yard stacking distribution* $(x_1, x_2, \cdots, x_{Num})$*, in which* x_τ *represents the number of containers in the* τ^{th} *tier of the yard space.*

Proof of Lemma 2. (i) Assuming that the container in the lowest tier of each yard-stack would be retrieved first, the number of yard-stacks needed for storing rehandled containers is maximized. A yard-stack is completely empty after the container in the lowest tier of this yard-stack is retrieved. The empty yard-stack can be used for storing rehandled containers in the subsequent operation.

φ_m^τ-stacks refer to the yard-stacks including τ m-containers. When retrieving m-containers from φ_m^τ-stacks, the number of yard-stacks needed for storing rehandled containers is $(\tau-1)\varphi_m^\tau - (\varphi_m^\tau - 1)$, in which $\varphi_m^\tau - 1$ counts the number of empty yard-stacks generated in the retrieving process, and these yard-stacks can be used for the next rehandling. Thus, the total number of yard-stacks needed for storing rehandled containers is $\sum_{\tau=2}^{Num}[(\tau-2)\varphi_m^\tau + 1] - [\sum_{\tau=2}^{Num}\varpi(\varphi_m^\tau) - 1]$. Here, $\sum_{\tau=2}^{Num}\varpi(\varphi_m^\tau) - 1$ counts the number of empty yard-stacks after clearing containers out from φ_m^τ-stacks ($\forall\tau$), and the number of empty yard-stacks must be eliminated from $\sum_{\tau=2}^{Num}[(\tau-2)\varphi_m^\tau + 1]$.

(ii) Fig. B.1 shows the changing process of a stacking distribution when retrieving a container from the yard. Assume that the number of m-containers in a yard-stack is no more than three, namely, there are α_3, β_3 and γ_3 containers in the first, second and third tiers, respectively. Six possible distributions may be produced when only retrieving a container from distribution $(\alpha_3,\ \beta_3,\ \gamma_3)$: (1) If the retrieved container is in the first tier and there are no containers stacked on top of it, then the distribution becomes $(\alpha_3 - 1, \beta_3, \gamma_3)$, and no rehandling is needed. (2) If the retrieved container is in the second tier and there are no containers stacked on top of it, then the distribution becomes $(\alpha_3, \beta_3 - 1, \gamma_3)$ and no rehandling is needed. (3) If the retrieved container is in the third tier and there are no containers stacked on top of it, then the distribution becomes $(\alpha_3, \beta_3, \gamma_3 - 1)$ and no rehandle is needed. (4) If the retrieved container is in the first tier of a two-tier stack, namely, there is a container stacked on top of it, then the distribution becomes $(\alpha_3, \beta_3 - 1, \gamma_3)$ and one rehandle is needed. (5) If the retrieved container is in the first tier of a three-tier stack, namely, there are two containers stacked on top of it, then the distribution becomes $(\alpha_3 + 1, \beta_3 - 1, \gamma_3 - 1)$ and two rehandles are required. (6) If the retrieved container is in the second tier of a three-tier stack, namely, there is a container stacked on top of it, then the distribution becomes $(\alpha_3 + 1, \beta_3 - 1, \gamma_3 - 1)$ and one rehandle is needed.

Hence, the iterative formulation for the expected number of rehandles can be expressed as

$$\begin{aligned} R(\alpha_3, \beta_3, \gamma_3) = {} & \frac{\alpha_3 - \beta_3}{\alpha_3 + \beta_3 + \gamma_3}[0 + R(\alpha_3 - 1, \beta_3, \gamma_3)] + \frac{\beta_3 - \gamma_3}{\alpha_3 + \beta_3 + \gamma_3}[0 + R(\alpha_3, \beta_3 - 1, \gamma_3)] \\ & + \frac{\gamma_3}{\alpha_3 + \beta_3 + \gamma_3}[0 + R(\alpha_3, \beta_3, \gamma_3 - 1)] + \frac{\beta_3 - \gamma_3}{\alpha_3 + \beta_3 + \gamma_3}[1 + R(\alpha_3, \beta_3 - 1, \gamma_3)] \\ & + \frac{\gamma_3}{\alpha_3 + \beta_3 + \gamma_3}[2 + R(\alpha_3 + 1, \beta_3 - 1, \gamma_3 - 1)] + \frac{\gamma_3}{\alpha_3 + \beta_3 + \gamma_3}[1 + R(\alpha_3 + 1, \beta_3 - 1, \gamma_3 - 1)] \end{aligned}$$

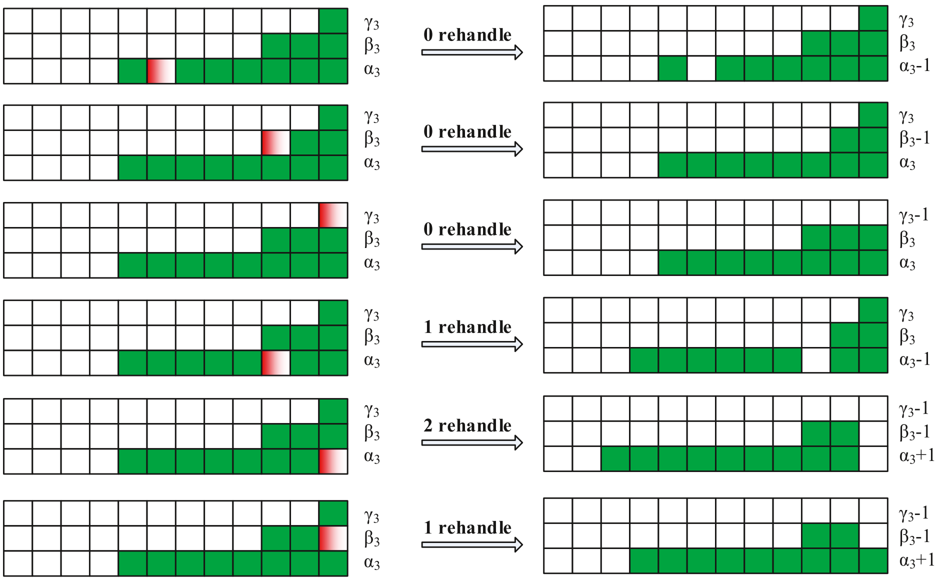

Fig. B.1. Stacking distribution changing process.

$$
\begin{aligned}
&= \frac{\alpha_3-\beta_3}{\alpha_3+\beta_3+\gamma_3}R(\alpha_3-1,\beta_3,\gamma_3)+\frac{2(\beta_3-\gamma_3)}{\alpha_3+\beta_3+\gamma_3}R(\alpha_3,\beta_3-1,\gamma_3)+\frac{\gamma_3}{\alpha_3+\beta_3+\gamma_3}R(\alpha_3,\beta_3,\gamma_3-1)\\
&\quad+\frac{2\gamma_3}{\alpha_3+\beta_3+\gamma_3}R(\alpha_3+1,\beta_3-1,\gamma_3-1)+\frac{\beta_3+2\gamma_3}{\alpha_3+\beta_3+\gamma_3}.
\end{aligned}
$$

Assuming that $R(\alpha_3,\beta_3,\gamma_3)=\frac{\beta_3}{2}+\frac{2\gamma_3}{3}$, we substitute $R(\alpha_3-1,\beta_3,\gamma_3)=\frac{\beta_3}{2}+\frac{2\gamma_3}{3}$, $R(\alpha_3,\beta_3-1,\gamma_3)=\frac{\beta_3-1}{2}+\frac{2\gamma_3}{3}$, $R(\alpha_3,\beta_3,\gamma_3-1)=\frac{\beta_3}{2}+\frac{2(\gamma_3-1)}{3}$ and $R(\alpha_3+1,\beta_3-1,\gamma_3-1)=\frac{\beta_3-1}{2}+\frac{2(\gamma_3-1)}{3}$ into the above formula. $\frac{\beta_3}{2}+\frac{2\gamma_3}{3}$ can be derived as the expected number of rehandles for the three-tier distribution.

In a similar manner, $R(\alpha_4,\beta_4,\gamma_4,\rho_4)=\frac{\beta_4}{2}+\frac{2\gamma_4}{3}+\frac{3\rho_4}{4}$ can be concluded for a four-tier distribution. Therefore, the general formulation for a τ-tier distribution can be given as $R(x_1,x_2,\cdots,x_{Num})=\sum_{\tau=2}^{Num}\frac{(\tau-1)x_\tau}{\tau}$. Generally, the allowed maximum height for yard-stacks is limited to 4 tiers, namely, *Num* is usually set as 4.

For $Num=4$,

$$
\begin{aligned}
R(x_1,x_2,x_3,x_4) &= \frac{x_1-x_2}{x_1+x_2+x_3+x_4}[0+R(x_1-1,x_2,x_3,x_4)]+\frac{x_2-x_3}{x_1+x_2+x_3+x_4}[0+R(x_1,x_2-1,x_3,x_4)]\\
&\quad+\frac{x_3-x_4}{x_1+x_2+x_3+x_4}[0+R(x_1,x_2,x_3-1,x_4)]+\frac{x_4}{x_1+x_2+x_3+x_4}[0+R(x_1,x_2,x_3,x_4-1)]\\
&\quad+\frac{x_2-x_3}{x_1+x_2+x_3+x_4}[1+R(x_1,x_2-1,x_3,x_4)]+\frac{x_3-x_4}{x_1+x_2+x_3+x_4}[1+R(x_1+1,x_2-1,x_3-1,x_4)]\\
&\quad+\frac{x_4}{x_1+x_2+x_3+x_4}[1+R(x_1+1,x_2,x_3-1,x_4-1)]\\
&\quad+\frac{x_3-x_4}{x_1+x_2+x_3+x_4}[2+R(x_1+1,x_2-1,x_3-1,x_4)]\\
&\quad+\frac{x_4}{x_1+x_2+x_3+x_4}[2+R(x_1+2,x_2-1,x_3-1,x_4-1)]\\
&\quad+\frac{x_4}{x_1+x_2+x_3+x_4}[3+R(x_1+2,x_2-1,x_3-1,x_4-1)]\\
&= \frac{x_1-x_2}{x_1+x_2+x_3+x_4}R(x_1-1,x_2,x_3,x_4)+\frac{2(x_2-x_3)}{x_1+x_2+x_3+x_4}R(x_1,x_2-1,x_3,x_4)\\
&\quad+\frac{x_3-x_4}{x_1+x_2+x_3+x_4}R(x_1,x_2,x_3-1,x_4)+\frac{x_4}{x_1+x_2+x_3+x_4}R(x_1,x_2,x_3,x_4-1)\\
&\quad+\frac{2(x_3-x_4)}{x_1+x_2+x_3+x_4}R(x_1+1,x_2-1,x_3-1,x_4)+\frac{x_4}{x_1+x_2+x_3+x_4}R(x_1+1,x_2,x_3-1,x_4-1)\\
&\quad+\frac{2x_4}{x_1+x_2+x_3+x_4}R(x_1+2,x_2-1,x_3-1,x_4-1)+\frac{x_2+2x_3+3x_4}{x_1+x_2+x_3+x_4}.
\end{aligned}
$$

Assuming that $R(x_1, x_2, x_3, x_4) = \frac{1}{2}x_2 + \frac{2}{3}x_3 + \frac{3}{4}x_4$, we substitute $R(x_1 - 1, x_2, x_3, x_4) = \frac{1}{2}x_2 + \frac{2}{3}x_3 + \frac{3}{4}x_4$, $R(x_1, x_2 - 1, x_3, x_4) = \frac{1}{2}(x_2 - 1) + \frac{2}{3}x_3 + \frac{3}{4}x_4$, $R(x_1, x_2, x_3 - 1, x_4) = \frac{1}{2}x_2 + \frac{2}{3}(x_3 - 1) + \frac{3}{4}x_4$, $R(x_1, x_2, x_3, x_4 - 1) = \frac{1}{2}x_2 + \frac{2}{3}x_3 + \frac{3}{4}(x_4 - 1)$, $R(x_1 + 1, x_2 - 1, x_3 - 1, x_4) = \frac{1}{2}(x_2 - 1) + \frac{2}{3}(x_3 - 1) + \frac{3}{4}x_4$, $R(x_1 + 1, x_2, x_3 - 1, x_4 - 1) = \frac{1}{2}x_2 + \frac{2}{3}(x_3 - 1) + \frac{3}{4}(x_4 - 1)$ and $R(x_1 + 2, x_2 - 1, x_3 - 1, x_4 - 1) = \frac{1}{2}(x_2 - 1) + \frac{2}{3}(x_3 - 1) + \frac{3}{4}(x_4 - 1)$ into the above formula. $\frac{1}{2}x_2 + \frac{2}{3}x_3 + \frac{3}{4}x_4$ can be derived as the expected number of rehandles for the four-tier distribution.

In general,

$$\begin{aligned}
R(x_1, x_2, \ldots, x_{Num}) = {} & \frac{x_1 - x_2}{\sum_{\tau=1}^{Num} x_\tau} R(x_1 - 1, x_2, \ldots, x_{Num}) + \frac{x_2 - x_3}{\sum_{\tau=1}^{Num} x_\tau} R(x_1, x_2 - 1, \ldots, x_{Num}) + \ldots \\
& + \frac{x_{Num-1} - x_{Num}}{\sum_{\tau=1}^{Num} x_\tau} R(x_1, x_2, \ldots, x_{Num-1} - 1, x_{Num}) + \frac{x_{Num}}{\sum_{\tau=1}^{Num} x_\tau} R(x_1, x_2, \ldots, x_{Num-1}, x_{Num} - 1) \\
& + \frac{x_2 - x_3}{\sum_{\tau=1}^{Num} x_\tau} [1 + R(x_1, x_2 - 1, \ldots, x_{Num})] + \frac{x_3 - x_4}{\sum_{\tau=1}^{Num} x_\tau} [1 + R(x_1 + 1, x_2 - 1, x_3 - 1, \ldots, x_{Num})] + \ldots \\
& + \frac{x_{Num-1} - x_{Num}}{\sum_{\tau=1}^{Num} x_\tau} [1 + R(x_1 + 1, x_2, \ldots, x_{Num-2} - 1, x_{Num-1} - 1, x_{Num}) \\
& + \frac{x_{Num}}{\sum_{\tau=1}^{Num} x_\tau} [1 + R(x_1 + 1, x_2, \ldots, x_{Num-2}, x_{Num-1} - 1, x_{Num} - 1)] \\
& + \frac{x_3 - x_4}{\sum_{\tau=1}^{Num} x_\tau} [2 + R(x_1 + 1, x_2 - 1, x_3 - 1, \ldots, x_{Num})] \\
& + \frac{x_4 - x_5}{\sum_{\tau=1}^{Num} x_\tau} [2 + R(x_1 + 2, x_2 - 1, x_3 - 1, x_4 - 1 \ldots, x_{Num})] \\
& + \frac{x_{Num}}{\sum_{\tau=1}^{Num} x_\tau} [2 + R(x_1 + 2, x_2, \ldots, x_{Num-2} - 1, x_{Num-1} - 1, x_{Num} - 1)] + \ldots \\
& + \frac{x_{Num}}{\sum_{\tau=1}^{Num} x_\tau} [Num - 1 + R(x_1 + Num - 2, x_2 - 1, \ldots, x_{Num} - 1)].
\end{aligned}$$

Assuming that $R(x_1, x_2, \ldots, x_{Num}) = \sum_{\tau=2}^{Num} \frac{(\tau-1)x_\tau}{\tau}$ and setting $F = \sum_{\tau=2}^{Num} \frac{(\tau-1)x_\tau}{\tau}$, we have $R(x_1, x_2 - 1, \ldots, x_\tau) = F - \frac{1}{2}$, $R(x_1, x_2, x_3 - 1, \ldots, x_{Num}) = F - \frac{2}{3}$, ..., $R(x_1 + Num - 2, x_2 - 1, \ldots, x_{Num} - 1) = F - \frac{1}{2} - \frac{2}{3} - \ldots - \frac{Num-1}{Num}$. Substituting them into the above formula, we can obtain

$$\begin{aligned}
R(x_1, x_2, \ldots, x_{Num}) = {} & \frac{x_1 - x_2}{\sum_{\tau=1}^{Num} x_\tau} F + \frac{x_2 - x_3}{\sum_{\tau=1}^{Num} x_\tau} \left(F - \frac{1}{2}\right) + \ldots + \frac{x_{Num-1} - x_{Num}}{\sum_{\tau=1}^{Num} x_\tau} \left(F - \frac{Num-2}{Num-1}\right) \\
& + \frac{x_{Num}}{\sum_{\tau=1}^{Num} x_\tau} \left(F - \frac{Num-1}{Num}\right) + \frac{x_2 - x_3}{\sum_{\tau=1}^{Num} x_\tau} \left[1 + F - \frac{1}{2}\right] \\
& + \frac{x_3 - x_4}{\sum_{\tau=1}^{Num} x_\tau} \left[1 + F - \frac{1}{2} - \frac{2}{3}\right] + \ldots + \frac{x_{Num-1} - x_{Num}}{\sum_{\tau=1}^{Num} x_\tau} \left[1 + F - \frac{Num-3}{Num-2} - \frac{Num-2}{Num-1}\right) \\
& + \frac{x_{Num}}{\sum_{\tau=1}^{Num} x_\tau} \left[1 + F - \frac{Num-2}{Num-1} - \frac{Num-1}{Num}\right] + \frac{x_3 - x_4}{\sum_{\tau=1}^{Num} x_\tau} \left[2 + F - \frac{1}{2} - \frac{2}{3}\right] \\
& + \frac{x_4 - x_5}{\sum_{\tau=1}^{Num} x_\tau} \left[2 + F - \frac{1}{2} - \frac{2}{3} - \frac{3}{4}\right] + \ldots \\
& + \frac{x_{Num}}{\sum_{\tau=1}^{Num} x_\tau} \left[2 + F - \frac{Num-3}{Num-2} - \frac{Num-2}{Num-1} - \frac{Num-1}{Num}\right] + \ldots \\
& + \frac{x_{Num}}{\sum_{\tau=1}^{Num} x_\tau} \left[Num - 1 + F - \frac{1}{2} - \frac{2}{3} - \ldots - \frac{Num-1}{Num}\right] \\
= {} & \left(\frac{x_1 - x_2}{\sum_{\tau=1}^{Num} x_\tau} + \frac{2(x_2 - x_3)}{\sum_{\tau=1}^{Num} x_\tau} + \ldots + \frac{Num * x_{Num}}{\sum_{\tau=1}^{Num} x_\tau}\right) F + \left(-\frac{1}{2} - \frac{1}{2} + 1\right) \frac{x_2 - x_3}{\sum_{\tau=1}^{Num} x_\tau} \\
& + \left(-\frac{2}{3} + 1 - \frac{1}{2} - \frac{2}{3} + 2 - \frac{1}{2} - \frac{2}{3}\right) \frac{x_3 - x_4}{\sum_{\tau=1}^{Num} x_\tau} + \ldots = F.
\end{aligned}$$

Thus, the assumption is proved.

The formulation demonstrates the relation between the expected number of rehandles and the stacking height of a container group in yard-stacks. It indicates that each group of containers should be distributed to the lower tier of the available storage space to achieve the optimal distribution. □

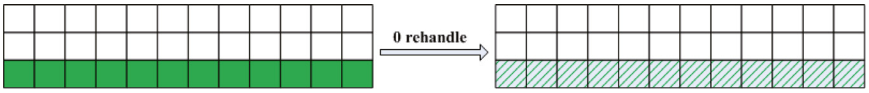

Fig. B.2. Containers distributed in one tier of the yard space (the one-tier distribution).

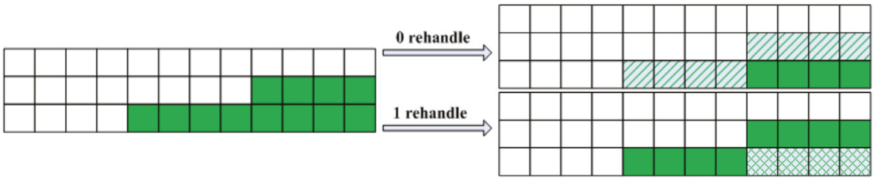

Fig. B.3. Containers distributed in two tiers of the yard space (the two-tier distribution).

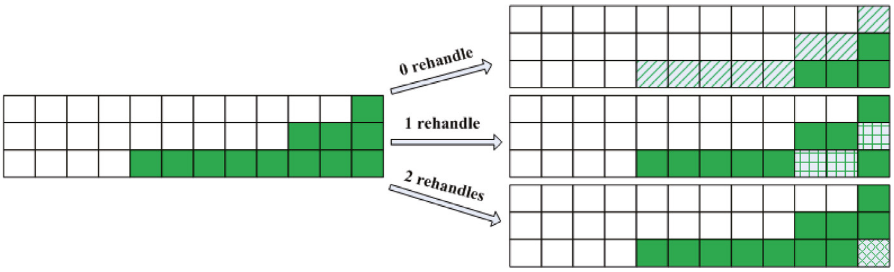

Fig. B.4. Containers distributed in three tiers of the yard space (the three-tier distribution).

Lemma 3. *If some containers are distributed into one tier, two tiers or three tiers of the yard space, and the number of containers in the ground tier of the two-tier distribution equals that of the three-tier distribution, then the expected number of rehandles of the three-tier distribution is the highest, and the expected number of rehandles of the one-tier distribution is lower than that of the two-tier distribution*

Proof of Lemma 3. As illustrated in Figs. B.2–B.4, 12 containers are distributed across one, two or three tiers of the yard space, respectively. The one-tier distribution in Fig. B.2 is the optimal distribution of 12 containers, while the two-tier distribution in Fig. B.3 causes four rehandles at most if retrieving the four containers from the branch of “1 rehandle” in the shadow first. Similarly, the three-tier distribution in Fig. B.4 yields four rehandles if retrieving the container in the shadow from the branch of “2 rehandles” first and then retrieving the two containers in the shadow from the branch of “1 rehandle” second. Assume that the empty yard-stack has the first priority for storing rehandled containers. Before retrieving the marked container in the shadow from the branch of “2 rehandles”, the marked container in the last yard-stack in the branch of “1 rehandle” is rehandled to an empty yard-stack first. Concerning the two-tier distribution in Fig. B.3 and the three-tier distribution in Fig. B.4, the pessimistic number of rehandles is the same if there are sufficient yard-stacks for storing rehandled containers, as proved in Lemma 2(i). However, the expected number of rehandles for the two-tier distribution is less than that for the three-tier distribution, which is proved in the following.

The one-tier-distribution is denoted by (α_0), the two-tier-distribution is denoted by (α_1, β_1), and the three-tier-distribution is denoted by $(\alpha_2, \beta_2, \gamma_2)$. As $R(\alpha_0) = 0$, $R(\alpha_1, \beta_1) > 0$, $R(\alpha_2, \beta_2, \gamma_2) > 0$, it is obvious that $R(\alpha_0)$ is the minimum value, but it is not sure which is the maximum value.

- Assume that the containers would be rehandled once at most.

According to Lemma 2(ii), we can find that $R(\alpha_1, \beta_1) = \frac{\beta_1}{2}$, and $R(\alpha_2, \beta_2, \gamma_2) = \frac{\beta_2}{2} + \frac{2\gamma_2}{3}$. For the two distributions in Figs. B.3 and B.4, $\alpha_1 + \beta_1 = \alpha_2 + \beta_2 + \gamma_2$, $\alpha_1 = \alpha_2$, and $\beta_1 = \beta_2 + \gamma_2$. Thus, $R(\alpha_1, \beta_1) = \frac{\beta_2+\gamma_2}{2}$, while $R(\alpha_2, \beta_2, \gamma_2) = \frac{\beta_2+\gamma_2}{2} + \frac{\gamma_2}{6}$. Thus, $R(\alpha_2, \beta_2, \gamma_2) > R(\alpha_1, \beta_1)$ is true.

To conclude, the two-tier distribution in Fig. B.3 is better than the three-tier distribution in Fig. B.4. However, the one-tier distribution in Fig. B.2 is not easy to achieve based on the ship stowage plan. Thus, the two-tier distribution in Fig. B.3 is easy to obtain compared with the one-tier distribution, and it is better than the three-tier distribution in Fig. B.4.

- Assume that containers may be rehandled twice, and there exists at least one container in each yard-stack initially.

$R(G, \beta_1)$ can be expressed as

$$R(G,\beta_1)=\frac{\beta_1(G+2)}{2(G+1)}=\frac{(\beta_2+\gamma_2)\cdot(G+2)}{2(G+1)}. \tag{B.1}$$

$R(G, \beta_2, \gamma_2)$ can be expressed as

$$\begin{aligned}R(G,\beta_2,\gamma_2)&=\frac{G-\beta_2}{G+\beta_2+\gamma_2}[0+R(G-1,\beta_2,\gamma_2)]+\frac{\beta_2-\gamma_2}{G+\beta_2+\gamma_2}[0+R(G,\beta_2-1,\gamma_2)]\\&\quad+\frac{\gamma_2}{G+\beta_2+\gamma_2}[0+R(G,\beta_2,\gamma_2-1)]+\frac{\beta_2-\gamma_2}{G+\beta_2+\gamma_2}[1+R(G-1,\beta_2,\gamma_2)]\\&\quad+\frac{\gamma_2}{G+\beta_2+\gamma_2}[1+R(G,\beta_2,\gamma_2-1)]+\frac{\gamma_2}{G+\beta_2+\gamma_2}[2+R(G-1,\beta_2+1,\gamma_2-1)]\\&=\frac{G-\gamma_2}{G+\beta_2+\gamma_2}R(G-1,\beta_2,\gamma_2)+\frac{\beta_2-\gamma_2}{G+\beta_2+\gamma_2}R(G,\beta_2-1,\gamma_2)+\frac{2\gamma_2}{G+\beta_2+\gamma_2}R(G,\beta_2,\gamma_2-1)\\&\quad+\frac{\gamma_2}{G+\beta_2+\gamma_2}R(G-1,\beta_2+1,\gamma_2-1)+\frac{\beta_2+2\gamma_2}{G+\beta_2+\gamma_2}.\end{aligned} \tag{B.2}$$

Assuming that

$$R(G,\beta_2,\gamma_2)>\frac{(\beta_2+\gamma_2)\cdot(G+2)}{2(G+1)}, \tag{B.3}$$

$R(G-1,\beta_2,\gamma_2)>\frac{\beta_2+\gamma_2}{2}$ can be concluded from the previous condition. Therefore,

$$\begin{aligned}R(G,\beta_2,\gamma_2)&>\frac{G-\gamma_2}{G+\beta_2+\gamma_2}\cdot\frac{\beta_2+\gamma_2}{2}+\frac{\beta_2-\gamma_2}{G+\beta_2+\gamma_2}\cdot\frac{(\beta_2+\gamma_2-1)\cdot(G+2)}{2(G+1)}\\&\quad+\frac{2\gamma_2}{G+\beta_2+\gamma_2}\cdot\frac{(\beta_2+\gamma_2-1)\cdot(G+2)}{2(G+1)}+\frac{\gamma_2}{G+\beta_2+\gamma_2}\cdot\frac{\beta_2+\gamma_2}{2}+\frac{\beta_2+2\gamma_2}{G+\beta_2+\gamma_2}.\end{aligned} \tag{B.4}$$

Namely,

$$R(G,\beta_2,\gamma_2)>\frac{G^2\beta_2+G(\beta_2)^2+G^2\gamma_2+G(\gamma_2)^2+2G\beta_2+4G\gamma_2+2\gamma_2+2G\beta_2\gamma_2+2(\beta_2)^2+2(\gamma_2)^2+4\beta_2\gamma_2}{2(G+1)\cdot(G+\beta_2+\gamma_2)}. \tag{B.5}$$

Besides,

$$\begin{aligned}\frac{(\beta_2+\gamma_2)\cdot(G+2)}{2(G+1)}&=\frac{G-\gamma_2}{G+\beta_2+\gamma_2}\cdot\frac{\beta_2+\gamma_2}{2}+\frac{\beta_2-\gamma_2}{G+\beta_2+\gamma_2}\cdot\frac{(\beta_2+\gamma_2-1)\cdot(G+2)}{2(G+1)}\\&\quad+\frac{2\gamma_2}{G+\beta_2+\gamma_2}\cdot\frac{(\beta_2+\gamma_2-1)\cdot(G+2)}{2(G+1)}+\frac{\gamma_2}{G+\beta_2+\gamma_2}\cdot\frac{\beta_2+\gamma_2}{2}+\frac{\beta_2+2\gamma_2}{G+\beta_2+\gamma_2}\\&=\frac{G^2\beta_2+G(\beta_2)^2+G^2\gamma_2+G(\gamma_2)^2+2G\beta_2+2G\gamma_2+2G\beta_2\gamma_2+2(\beta_2)^2+2(\gamma_2)^2+4\beta_2\gamma_2}{2(G+1)\cdot(G+\beta_2+\gamma_2)}.\end{aligned} \tag{B.6}$$

By (B.1), (B.5) and (B.6), it is obvious that $R(G, \beta_2, \gamma_2) > R(G, \beta_1)$; hence, assumption (B.3) is proved. □

Appendix C. Extension of Table C.1

Table C.1
(Extension) Computational results for the two methods on small-scale instances.

Instances	*C*	*G*	*Num*	*S*	*CPLEX*		*TS*	
					objective	$t(s)$	objective	$t(s)$
6-1	6	2	4	3	8	0.32	8	< 0.01
6-2	6	2	4	3	10	0.82	10	< 0.01
6-3	6	2	4	3	10	0.83	10	< 0.01
6-4	6	2	4	3	6	4.03	6	< 0.01
6-5	6	2	4	3	10	4.59	10	< 0.01
6-6	6	2	4	3	10	4.67	10	< 0.01
6-7	6	2	4	3	18	5.52	18	< 0.01
6-8	6	2	4	3	10	6.98	10	< 0.01
6-9	6	2	4	3	6	7.12	6	< 0.01
6-10	6	2	4	3	10	7.87	10	< 0.01
7-1	7	3	4	3	7	0.73	7	< 0.01
7-2	7	3	4	3	7	7.05	7	< 0.01
7-3	7	3	4	3	7	8.23	7	< 0.01
7-4	7	3	4	3	9	8.52	9	< 0.01
7-5	7	3	4	3	7	11.37	7	< 0.01
7-6	7	3	4	3	7	12.56	7	< 0.01
7-7	7	3	4	3	9	17.23	9	< 0.01
7-8	7	3	4	3	7	21.09	7	< 0.01
7-9	7	3	4	3	11	35.17	11	< 0.01
7-10	7	3	4	3	11	36.223	11	< 0.01
8-1	8	3	4	3	8	1.29	8	< 0.01
8-2	8	3	4	3	8	11.08	8	< 0.01
8-3	8	3	4	3	10	18.37	10	< 0.01
8-4	8	3	4	3	8	30.96	8	< 0.01
8-5	8	3	4	3	8	32.68	8	< 0.01
8-6	8	3	4	3	9	36.63	9	< 0.01
8-7	8	3	4	3	10	42.191	10	< 0.01
8-8	8	3	4	3	12	65.27	12	< 0.01
8-9	8	3	4	3	14	86.23	14	< 0.01
8-10	8	3	4	3	12	146.32	12	< 0.01
9-1	9	3	4	3	9	6.96	9	< 0.01
9-2	9	3	4	3	13	21.27	13	< 0.01
9-3	9	3	4	3	11	46.55	11	< 0.01
9-4	9	3	4	3	13	72.36	13	< 0.01
9-5	9	3	4	3	13	95.24	13	< 0.01
9-6	9	3	4	3	15	99.38	15	< 0.01
9-7	9	3	4	3	27	142.60	27	< 0.01
9-8	9	3	4	3	11	159.29	11	< 0.01
9-9	9	3	4	3	15	163.17	15	< 0.01
9-10	9	3	4	3	13	396.25	13	< 0.01
10-1	10	4	4	3	10	21.36	10	< 0.01
10-2	10	4	4	3	10	45.83	10	0.01
10-3	10	4	4	3	10	132.17	10	< 0.01
10-4	10	4	4	3	12	168.75	12	0.01
10-5	10	4	4	3	14	172.49	14	< 0.01
10-6	10	4	4	3	12	202.37	12	< 0.01
10-7	10	4	4	3	12	263.86	12	0.01
10-8	10	4	4	3	12	1266.29	12	0.01
10-9	10	4	4	3	12	1307.28	12	0.01
10-10	10	4	4	3	12	2437.58	12	0.01
11-1	11	4	4	3	15	110.63	15	< 0.01
11-2	11	4	4	3	13	138.27	13	< 0.01
11-3	11	4	4	3	15	192.76	15	< 0.01
11-4	11	4	4	3	11	326.90	11	< 0.01
11-5	11	4	4	3	15	467.59	15	0.01
11-6	11	4	4	3	13	506.23	13	< 0.01
11-7	11	4	4	3	13	1238.71	13	< 0.01
11-8	11	4	4	3	15	5209.29	15	0.01
11-9	11	4	4	3	15	8621.68	15	< 0.01
11-10	11	4	4	3	15	15231.23	15	< 0.01
12-1	12	4	4	3	12	162.73	12	0.01
12-2	12	4	4	3	14	253.92	14	< 0.01
12-3	12	4	4	3	8	839.69	8	0.01
12-4	12	4	4	3	42	957.27	42	0.01
12-5	12	4	4	3	22	1703.39	22	0.01
12-6	12	4	4	3	18	3299.76	18	0.01
12-7	12	4	4	3	16	4738.55	16	0.01
12-8	12	4	4	3	–	–	14	0.01
12-9	12	4	4	3	–	–	17	0.02
12-10	12	4	4	3	–	–	36	0.02

Appendix D. Instance for the initial stage

Fig. D.1 illustrates the decision process in *the initial stage*.

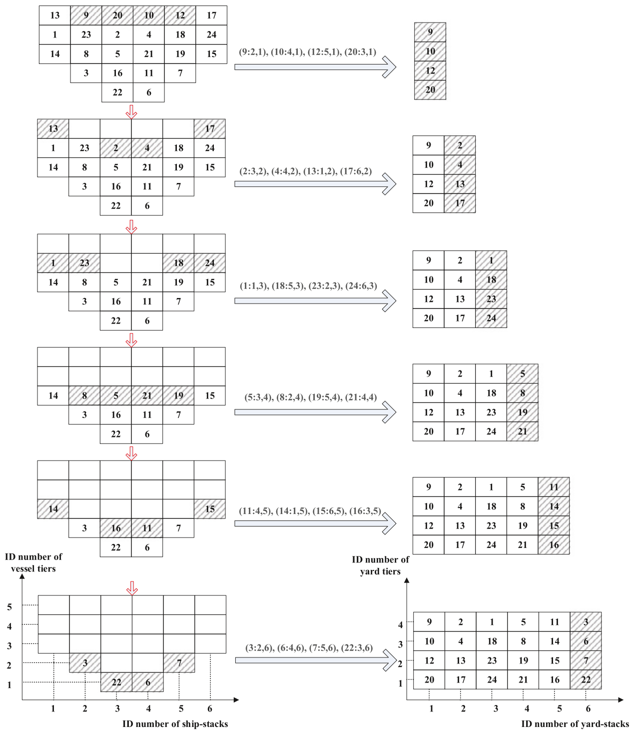

Fig. D.1. Illustration of an example for *the initial stage*.

Appendix E. Sensitivity analysis of the time window length

Table E.1
Expected number of rehandles required on four types of yard sizes and six of ship-bay sizes (12 time windows).

Ship-bay size	Yard size	Average expected number of rehandles for 100 instances									
$H \times S$	$Num \times G$	AT(s)	NS-0	NS-1	NS-2	NS-3	NS-4	NS-5	NS-6	NS-7	...
10 × 5	2 × 25	0.01	1.23	0.07	**0.02**	0.02	0.02	0.02	0.02	0.02	0.02
	3 × 17	0.01	1.86	0.30	**0.22**	0.22	0.22	0.22	0.22	0.22	0.22
	4 × 13	0.02	2.785	1.16	**1.15**	1.15	1.15	1.15	1.15	1.15	1.15
	5 × 10	0.03	4.00	**2.56**	2.56	2.56	2.56	2.56	2.56	2.56	2.56
20 × 5	2 × 50	0.02	2.14	0.07	**0.01**	0.01	0.01	0.01	0.01	0.01	0.01
	3 × 34	0.05	3.99	0.69	**0.57**	0.57	0.57	0.57	0.57	0.57	0.57
	4 × 25	0.09	5.94	2.51	**2.49**	2.49	2.49	2.49	2.49	2.49	2.49
	5 × 20	0.12	7.55	**4.74**	4.74	4.74	4.74	4.74	4.74	4.74	4.74
10 × 10	2 × 50	0.02	2.14	0.11	0.03	**0.01**	0.01	0.01	0.01	0.01	0.01
	3 × 34	0.02	3.89	0.37	0.06	**0.04**	0.04	0.04	0.04	0.04	0.04
	4 × 25	0.03	5.94	1.08	0.48	**0.47**	0.47	0.47	0.47	0.47	0.47
	5 × 20	0.05	8.06	2.44	1.73	**1.71**	1.71	1.71	1.71	1.71	1.71
20 × 10	2 × 100	0.05	4.31	0.11	0.03	0.02	**0.01**	0.01	0.01	0.01	0.01
	3 × 67	0.04	7.98	0.72	0.11	**0.04**	0.04	0.04	0.04	0.04	0.04
	4 × 50	0.11	12.08	2.10	0.87	**0.76**	0.76	0.76	0.76	0.76	0.76
	5 × 40	0.24	15.85	4.60	3.26	**3.25**	3.25	3.25	3.25	3.25	3.25
10 × 15	2 × 75	0.04	3.17	0.08	0.01	**0**	0	0	0	0	0
	3 × 50	0.03	5.95	0.51	0.07	0.01	**0**	0	0	0	0
	4 × 38	0.04	8.83	1.48	0.44	0.24	*0.20*	0.20	0.20	0.20	0.20
	5 × 30	0.07	11.79	3.24	1.36	1.02	**1.01**	1.01	1.01	1.01	1.01
20 × 15	2 × 150	0.10	6.12	0.22	**0**	0	0	0	0	0	0
	3 × 100	0.07	12.55	0.96	0.16	0.05	**0.02**	0.02	0.02	0.02	0.02
	4 × 75	0.10	17.89	3.04	0.65	0.28	**0.23**	0.23	0.23	0.23	0.23
	5 × 60	0.26	23.65	6.35	2.22	1.67	1.64	**1.63**	1.63	1.63	1.63

Table E.2
Expected number of rehandles required on four types of yard sizes and six of ship-bay sizes (10 time windows).

Ship-bay size	Yard size	Average expected number of rehandles for 100 instances									
$H\times S$	$Num\times G$	AT(s)	NS-0	NS-1	NS-2	NS-3	NS-4	NS-5	NS-6	NS-7	...
10 × 5	2 × 25	0.01	1.15	0.05	**0.03**	0.03	0.03	0.03	0.03	0.03	0.03
	3 × 17	0.01	2.42	0.53	**0.45**	0.45	0.45	0.45	0.45	0.45	0.45
	4 × 13	0.02	3.42	1.61	**1.59**	1.59	1.59	1.59	1.59	1.59	1.59
	5 × 10	0.03	4.63	**3.29**	3.29	3.29	3.29	3.29	3.29	3.29	3.29
20 × 5	2 × 50	0.02	2.56	0.16	0.03	**0.02**	0.02	0.02	0.02	0.02	0.02
	3 × 34	0.04	4.82	0.96	**0.82**	0.82	0.82	0.82	0.82	0.82	0.82
	4 × 25	0.09	6.98	3.33	**3.29**	3.29	3.29	3.29	3.29	3.29	3.29
	5 × 20	0.11	9.18	**6.15**	6.15	6.15	6.15	6.15	6.15	6.15	6.15
10 × 10	2 × 50	0.02	2.65	0.11	0.02	**0**	0	0	0	0	0
	3 × 34	0.02	4.95	0.54	0.12	**0.09**	0.09	0.09	0.09	0.09	0.09
	4 × 25	0.03	7.17	1.61	0.78	**0.71**	0.71	0.71	0.71	0.71	0.71
	5 × 20	0.06	9.33	3.53	2.72	**2.71**	2.71	2.71	2.71	2.71	2.71
20 × 10	2 × 100	0.04	4.96	0.25	0.02	**0**	0	0	0	0	0
	3 × 67	0.04	10.03	1.02	0.22	**0.13**	0.13	0.13	0.13	0.13	0.13
	4 × 50	0.13	14.30	3.41	1.61	**1.45**	1.45	1.45	1.45	1.45	1.45
	5 × 40	0.25	18.53	6.52	4.85	**4.84**	4.84	4.84	4.84	4.84	4.84
10 × 15	2 × 75	0.03	3.64	0.10	0.02	**0**	0	0	0	0	0
	3 × 50	0.02	7.38	0.85	0.16	0.05	**0.02**	0.02	0.02	0.02	0.02
	4 × 38	0.03	10.57	2.14	0.65	0.35	**0.31**	0.31	0.31	0.31	0.31
	5 × 30	0.07	14.27	4.82	2.34	1.83	**1.82**	1.82	1.82	1.82	1.82
20 × 15	2 × 150	0.08	7.23	0.27	0.02	**0**	0	0	0	0	0
	3 × 100	0.07	14.71	1.62	0.28	0.08	0.04	**0.03**	0.03	0.03	0.03
	4 × 75	0.13	21.55	4.48	1.42	0.67	0.59	**0.58**	0.58	0.58	0.58
	5 × 60	0.34	27.72	9.18	4.13	3.20	3.16	**3.15**	3.15	3.15	3.15

Table E.3
Expected number of rehandles required on four types of yard sizes and six of ship-bay sizes (6 time windows).

Ship-bay size	Yard size	Average expected number of rehandles for 100 instances									
$H\times S$	$Num\times G$	AT(s)	NS-0	NS-1	NS-2	NS-3	NS-4	NS-5	NS-6	NS-7	...
10 × 5	2 × 25	0.01	2.17	0.30	0.16	**0.15**	0.15	0.15	0.15	0.15	0.15
	3 × 17	0.02	4.04	1.24	**1.04**	1.04	1.04	1.04	1.04	1.04	1.04
	4 × 13	0.03	5.61	3.31	**3.28**	3.28	3.28	3.28	3.28	3.28	3.28
	5 × 10	0.03	7.34	**5.98**	5.98	5.98	5.98	5.98	5.98	5.98	5.98
20 × 5	2 × 50	0.02	4.12	0.43	**0.11**	0.11	0.11	0.11	0.11	0.11	0.11
	3 × 34	0.07	7.66	2.44	2.17	**2.16**	2.16	2.16	2.16	2.16	2.16
	4 × 25	0.12	11.57	7.12	**7.07**	7.07	7.07	7.07	7.07	7.07	7.07
	5 × 20	0.10	14.82	**11.90**	11.90	11.90	11.90	11.90	11.90	11.90	11.90
10 × 10	2 × 50	0.02	4.17	0.48	0.07	0.02	**0**	0	0	0	0
	3 × 34	0.03	7.87	2.02	0.80	0.54	**0.53**	0.53	0.53	0.53	0.53
	4 × 25	0.05	11.62	4.85	3.36	3.18	**3.17**	3.17	3.17	3.17	3.17
	5 × 20	0.07	15.02	8.54	7.52	**7.51**	7.51	7.51	7.51	7.51	7.51
20 × 10	2 × 100	0.05	8.12	1.01	0.17	0.04	**0.02**	0.02	0.02	0.02	0.02
	3 × 67	0.13	16.26	3.76	1.40	0.91	**0.90**	0.90	0.90	0.90	0.90
	4 × 50	0.27	23.14	9.65	6.55	**6.19**	6.19	6.19	6.19	6.19	6.19
	5 × 40	0.32	30.24	17.34	15.35	15.29	**15.27**	15.27	15.27	15.27	15.27
10 × 15	2 × 75	0.03	6.14	0.71	0.12	0.02	**0**	0	0	0	0
	3 × 50	0.04	12.14	3.07	1.13	0.51	0.32	**0.28**	0.28	0.28	0.28
	4 × 38	0.08	17.37	7.16	3.91	2.75	2.53	**2.52**	2.52	2.52	2.52
	5 × 30	0.12	21.98	12.35	8.73	8.05	8.02	**8.01**	8.01	8.01	8.01
20 × 15	2 × 150	0.08	12.49	1.35	0.23	0.02	0.01	**0**	0	0	0
	3 × 100	0.11	23.72	5.59	2.05	0.85	0.40	**0.31**	0.31	0.31	0.31
	4 × 75	0.37	34.62	13.98	7.42	4.94	4.58	4.55	**4.54**	4.54	4.54
	5 × 60	0.55	44.85	24.36	17.25	15.77	15.70	**15.69**	15.69	15.69	15.69

Table E.4
Expected number of rehandles required on four types of yard sizes and six of ship-bay sizes (4 time windows).

Ship-bay size	Yard size	Average expected number of rehandles for 100 instances									
$H \times S$	$Num \times G$	AT(s)	NS-0	NS-1	NS-2	NS-3	NS-4	NS-5	NS-6	NS-7	...
10 × 5	2 × 25	0.01	3.06	0.68	0.33	**0.32**	0.32	0.32	0.32	0.32	0.32
	3 × 17	0.03	5.63	2.76	**2.56**	2.56	2.56	2.56	2.56	2.56	2.56
	4 × 13	0.04	8.43	6.37	**6.36**	6.36	6.36	6.36	6.36	6.36	6.36
	5 × 10	0.03	10.53	**9.75**	9.75	9.75	9.75	9.75	9.75	9.75	9.75
20 × 5	2 × 50	0.05	6.05	1.28	0.63	**0.61**	0.61	0.61	0.61	0.61	0.61
	3 × 34	0.14	11.83	6.01	**5.53**	5.53	5.53	5.53	5.53	5.53	5.53
	4 × 25	0.14	16.79	13.13	**13.12**	13.12	13.12	13.12	13.12	13.12	13.12
	5 × 20	0.10	21.03	**19.30**	19.30	19.30	19.30	19.30	19.30	19.30	19.30
10 × 10	2 × 50	0.02	6.14	1.34	0.46	0.21	0.11	**0.09**	0.09	0.09	0.09
	3 × 34	0.07	11.43	5.04	3.10	2.61	**2.54**	2.54	2.54	2.54	2.54
	4 × 25	0.09	16.97	11.37	9.81	**9.58**	9.58	9.58	9.58	9.58	9.58
	5 × 20	0.08	20.95	17.28	**16.67**	16.67	16.67	16.67	16.67	16.67	16.67
20 × 10	2 × 100	0.08	12.53	2.73	0.89	0.39	0.19	**0.18**	0.18	0.18	0.18
	3 × 67	0.32	23.64	10.18	6.25	5.14	5.06	5.05	**5.04**	5.04	5.04
	4 × 50	0.40	33.42	21.96	18.47	**18.10**	18.10	18.10	18.10	18.10	18.10
	5 × 40	0.34	42.05	34.93	**33.79**	33.79	33.79	33.79	33.79	33.79	33.79
10 × 15	2 × 75	0.04	9.55	1.97	0.72	0.30	0.16	0.09	0.06	**0.04**	0.04
	3 × 50	0.11	17.39	7.68	4.72	3.37	2.57	**2.34**	2.34	2.34	2.34
	4 × 38	0.16	24.83	16.17	12.44	10.99	10.62	10.60	**10.59**	10.59	10.59
	5 × 30	0.16	31.19	26.09	24.03	**23.46**	23.46	23.46	23.46	23.46	23.46
20 × 15	2 × 150	0.12	18.83	3.73	1.18	0.47	0.16	0.07	0.04	**0.02**	0.02
	3 × 100	0.54	35.80	15.71	9.53	6.66	5.01	4.61	**4.59**	4.59	4.59
	4 × 75	0.76	50.67	32.11	25.37	22.41	21.78	21.73	**21.72**	21.72	21.72
	5 × 60	0.71	62.85	52.42	48.17	**47.32**	47.32	47.32	47.32	47.32	47.32

Appendix F. Constructive algorithm

We also designed a constructive algorithm (**CA**) for the ICUSP. Here are the definitions of notations in the *CA:* ***Ship*** represents the ship-bay distribution; ***Yard*** represents the planned yard stacking distribution; $Ship_0$ is the instant stacking status of ship-bay in the unloading process; $Yard_0$ is the instant stacking status of yard-stack in the storage process; *top* is defined as the set of containers that are in the top tier of each ship-stack in the unloading process; *bottom* is defined as the set of containers that are available in each yard-stack for storing from the ship; *tier* represents the number of containers instantly stacking in each yard-stack; and *label* records the largest retrieval priority of each yard-stack according to the stacked containers.

Step 1: Not considering the containers' distribution ***Ship*** in the ship-bay, generate the optimal stacking distribution based on Lemma 1, denoted by ***Yard***. Initialize $Ship_0$, $Yard_0$, *top, bottom, tier*, and *label*.

Step2: Unload containers from the ship-bay according to ***Ship***, and stack them in the yard in accordance with ***Yard*** if $top \bigcap bottom$ is not empty, and update $Ship_0$, $Yard_0$, *top, bottom, tier*, and *label*.

Step 3: If $top \bigcap bottom$ is empty, ***Yard*** would be adjusted.

Step 3a: Find and select a container prioritized by *min{bottom}* from *bottom*, and a container prioritized by *max{top}* from the other yard-stack, and exchange them without causing necessary and expected rehandles in both yard-stacks. If these two containers can be found, update ***Yard*** and *bottom*, and go back to Step 2.

Step 3b: Find and select a container prioritized *min{bottom}* from *bottom*, and a container prioritized *max{top}* from the other yard-stack, and exchange them without causing necessary rehandles in both yard-stacks. If these two containers can be found, update ***Yard*** and *bottom*, and go back to Step 2.

Step 3c: Clear containers which haven't stacked in the yard from ***Yard***, and reorganize them to the empty yard slot. To generate the new stacking distribution, first assign one of *max{top}* to the yard, and then assign the unloaded containers by the sequence of increasing retrieval priority to the optimal yard-stack. Update ***Yard*** and *bottom*, and go back to Step 2.

Numerical experiment is conducted on the efficiency of the *CA* and *TS* for ship-bay 20×15, therein involving four types of yard sizes by considering 4, 6, 8, 10 and 12 time windows. A total of 100 different instances are generated for each time window scenario. Table F.1 shows the performances of two algorithms both designed for the ICUSP. The columns marked by rehandles and AT record the average number of rehandles and runtime of 100 instances. The results show that the *TS* performs better in finding the optimal stacking distribution than the *CA* in any instances.

CA is designed based on Lemma 1 by pre-generating optimal stacking distribution without considering containers initial locations in the ship-bay. And the stacking distribution is constantly adjusted with containers unloading from the ship-bay and stacking into the yard space one by one. To generate a feasible stacking distribution, Step 3c in the *CA* allows the container with lower retrieval priority stacking on the container with higher retrieval priority. When the calculation enters Step 3c, *CA* cannot avoid the necessary rehandles, causing the higher number of rehandles.

Table F.1
Results of the instance comparison using the *CA* and *TS*.

Time windows	Yard size	CA		TS	
	$Num \times G$	rehandles	AT(s)	rehandles	AT(s)
4	2 × 150	0.02	0.02	0.02	0.13
	3 × 100	2.87	0.13	2.61	0.40
	4 × 75	19.37	0.12	19.12	0.68
	5 × 60	45.35	0.35	44.77	0.67
6	2 × 150	0.01	0.02	0	0.11
	3 × 100	0.60	0.09	0.09	0.10
	4 × 75	3.16	0.25	2.86	0.35
	5 × 60	12.65	0.20	11.57	0.55
8	2 × 150	0.09	0.02	0	0.10
	3 × 100	0.74	0.12	0.01	0.08
	4 × 75	2.62	0.25	0.58	0.15
	5 × 60	5.81	0.43	3.96	0.38
10	2 × 150	0.12	0.03	0	0.09
	3 × 100	0.92	0.15	0	0.07
	4 × 75	2.28	0.31	0.12	0.09
	5 × 60	3.81	0.41	1.59	0.26
12	2 × 150	0.16	0.05	0	0.09
	3 × 100	1.26	0.17	0	0.06
	4 × 75	2.45	0.34	0.10	0.08
	5 × 60	3.17	0.49	0.81	0.15

References

[1] I. Vacca, M. Salani, M. Bierlaire, An exact algorithm for the integrated planning of berth allocation and quay crane assignment, Transp. Sci. 47 (2) (2013) 148–161.
[2] J. Lehnfeld, S. Knust, Loading, unloading and premarshalling of stacks in storage areas: survey and classification, Eur. J. Oper. Res. 239 (2) (2014) 297–312.
[3] I.F.A. Vis, K.J. Roodbergen, Scheduling of container storage and retrieval, Oper. Res. 57 (2) (2009) 456–467.
[4] J.G. Jin, D. Lee, J.X. Cao, Storage yard management in maritime container terminals, Transp. Sci. 50 (4) (2014) 1300–1313.
[5] H. Rashidi, E.P.K. Tsang, Novel constraints satisfaction models for optimization problems in container terminals, Appl. Math. Model. 37 (6) (2013) 3601–3634.
[6] R. Stahlbock, S. Voss, Operations research at container terminals: a literature update, OR. Spectrum. 30 (1) (2008) 1–52.
[7] K.H. Kim, Evaluation of the number of rehandles in container yards, Comput. Oper. Res. 32 (4) (1997) 701–711.
[8] K.H. Kim, Y.M. Park, K.R. Ryu, Deriving decision rules to locate export containers in container yards, Eur. J. Oper. Res. 124 (1) (2000) 89–101.
[9] A. Imai, K. Sasaki, E. Nishimura, S. Papadimitriou, Multi-objective simultaneous stowage and load planning for a container ship with container rehandle in yard-stacks, Eur. J. Oper. Res. 171 (2) (2006) 373–389.
[10] K.H. Kim, G.P. Hong, A heuristic rule for relocating blocks, Comput. Oper. Res. 33 (4) (2006) 940–954.
[11] F. Forster, A. Bortfeldt, A tree search procedure for the container relocation problem, Comput. Oper. Res. 39 (2) (2012) 299–309.
[12] M.E.H. Petering, M.I. Hussein, A new mixed integer program and extended look-ahead heuristic algorithm for the block relocation problem, Eur. J. Oper. Res. 231 (1) (2013) 120–130.
[13] B. Jin, W. Zhu, A. Lim, Solving the container relocation problem by an improved greedy look-ahead heuristic, Eur. J. Oper. Res. 240 (3) (2015) 837–847.
[14] B. Jin, A. Lim, W. Zhu, A greedy look-ahead heuristic for the container relocation problem, in: Recent Trends in Applied Artificial Intelligence, Springer, Berlin, Heidelberg, 2013, pp. 181–190.
[15] Y. Lee, Y.J. Lee, A heuristic for retrieving containers from a yard, Comput. Oper. Res. 37 (6) (2010) 1139–1147.
[16] Y. Lee, N.Y. Hsu, An optimization model for the container pre-marshalling problem, Comput. Oper. Res. 34 (11) (2007) 3295–3313.
[17] M. Caserta, S. Voss, A corridor algorithm-based algorithm for the pre-marshalling problem, in: Applications of Evolutionary Computing, Springer, Berlin, Heidelberg, 2009, pp. 788–797.
[18] Y. Lee, S.L. Chao, A neighborhood search heuristic for pre-marshalling export containers, Eur. J. Oper. Res. 196 (2) (2009) 468–475.
[19] C. Expósito-Izquierdo, B. Melián-Batista, M. Moreno-Vega, Premarshalling problem: heuristic solution algorithm and instances generator, Expert. Syst. Appl. 39 (9) (2012) 8337–8349.
[20] A. Bortfeldt, F. Forster, A tree search procedure for the container pre-marshalling problem, Eur. J. Oper. Res. 217 (3) (2012) 531–540.
[21] S.H. Huang, T.H. Lin, Heuristic algorithms for container pre-marshalling problems, Comput. Ind. Eng. 62 (1) (2012) 13–20.
[22] N. Wang, B. Jin, A. Lim, Target-guided algorithms for the container pre-marshalling problem, Omega 53 (2015) 67–77.
[23] K.H. Kim, H.B. Kim, The optimal determination of the space requirement and the number of transfer cranes for import containers, Comput. Ind. Eng. 35 (3) (1998) 427–430.
[24] E. Guldogan, Simulation-based analysis for hierarchical storage assignment policies in a container terminal, Simul.-Trans. Soc. Mod. Simul. 87 (6) (2011) 523–537.
[25] T. Park, R. Choe, Y.H. Kim, K.R. Ryu, Dynamic adjustment of container stacking policy in an automated container terminal, Int. J. Prod. Econ. 133 (1) (2011) 385–392.
[26] J. Woo, K.H. Kim, Estimating the space requirement for outbound container inventories in port container terminals, Int. J. Prod. Econ. 133 (1) (2011) 293–301.
[27] M. Yu, X. Qi, Storage space allocation models for inbound containers in an automatic container terminal, Eur. J. Oper. Res. 226 (1) (2013) 32–45.
[28] W.C. Ng, K.L. Mak, M.K. Li, Yard planning for vessel services with a cyclical calling pattern, Eng. Optim. 42 (11) (2010) 1039–1054.
[29] C. Zhang, W. Chen, L. Shi, et al., A note on deriving decision rules to locate export containers in container yards, Eur. J. Oper. Res. 205 (2) (2010) 483–485.
[30] C. Zhang, T. Wu, K.H. Kim, et al., Conservative allocation models for outbound containers in container terminals, Eur. J. Oper. Res. 238 (1) (2014) 155–165.
[31] H.J. Carlo, I.F.A. Vis, K.J. Roodbergen, Storage yard operations in container terminals: literature overview, trends, and research directions, Eur. J. Oper. Res. 235 (2) (2014) 412–430.
[32] B. Borgman, E.V. Asperen, R. Dekker, Online rules for container stacking, OR Spectrum 32 (3) (2010) 687–716.
[33] W.C. Ng, K.L. Mak, Yard crane scheduling in port container terminals, App. Math. Model. 29 (3) (2005) 263–276.
[34] W.C. Ng, K.L. Mak, M.K. Li, Yard planning for vessel services with a cyclical calling pattern, Eng. Optim. 42 (11) (2010) 1039–1054.