European Journal of Operational Research 221 (2012) 64–73

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

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

Production, Manufacturing and Logistics

A container yard storage strategy for improving land utilization and operation efficiency in a transshipment hub port

Xinjia Jiang [a], Loo Hay Lee [a,*], Ek Peng Chew [a], Yongbin Han [a], Kok Choon Tan [b]

[a] Department of Industrial and Systems Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore
[b] Department of Decision Sciences, NUS Business School, National University of Singapore, 15 Kent Ridge Drive, Singapore 119245, Singapore

ARTICLE INFO

Article history:
Received 13 May 2011
Accepted 5 March 2012
Available online 10 March 2012

Keywords:
Logistics
Port operation
Space-sharing yard template
Consignment strategy
High-low workload balancing protocol

ABSTRACT

This paper studies the storage yard management problem in a busy transshipment hub, where intense loading and unloading activities have to be considered at the same time. The need to handle huge volumes of container traffic and the scarcity of land in the container port area pose serious challenges for the port operator to provide efficient services. A consignment strategy with a static yard template has been used to reduce the level of reshuffles in the yard, but it sacrifices on land utilization because of exclusive storage space reservation. Two space-sharing approaches are proposed to improve on the land utilization through dynamic reservation of storage space for different vessels during different shifts. Meanwhile, workload assignment among reserved spaces will also satisfy the high-low workload balancing protocol to reduce traffic congestion in the yard. A framework which integrates space reservation and workload assignment is proposed. Experimental results show that the framework is able to provide solutions for containers handling within much less storage space, while guarantee the least yard crane deployment.

1. Introduction

With the development of international trade, container terminals play a more important role in the world sea cargo transportation system. According to Drewry Shipping Consultants, the annual container traffic has increased more than five times from 87,947 million TEUs in 1990 to 463,634 million TEUs in 2009. The growth of annual transshipment traffic is even faster, at 8.4 times, from 15,479 million TEUs in 1990 to 130,572 million TEUs in 2009. This trend is expected to continue. The arrival of mega container vessels coupled with the scarcity of land pose great challenges of providing efficient services, especially for transshipment ports. Terminal operators have focused much attention on improving quay-side efficiency to shorten the processing time to turn around the vessel. However, overall terminal productivity will not benefit much from faster quay-side operations without effective storage yard strategies to manage the storage and retrieval of containers. The importance of storage yard planning for container terminals has also been highlighted in Vis and de Koster (2003), Steenken et al. (2004), Stahlbock and Voβ (2008), and Ku et al. (2010).

The handling effort of each container for yard planning consists of two parts; namely, moving between yard and quay by transporters and repositioning in the stack by yard cranes. When the container traffic is heavy and land available is scarce, multi-level stacking is a common practice in storage yard planning. This storage strategy is not only the potential cause of unproductive reshuffles of containers, but also results in high concentration of activities within a small area which may cause traffic congestion of transporters if activities in the yard are not properly coordinated. Many studies have been reported to remedy these two kinds of bad effects, both of which greatly affect the operational efficiency of the storage yard and thus the whole terminal.

To study the level of reshuffles, De Castilho and Daganzo (1993) present methods for measuring the amount of handling effort required which depend on stack heights and the adopted storage strategies for import containers. Kim (1997) proposes a methodology to estimate the expected number of reshuffles to pick up an arbitrary container and the total number of reshuffles to pick up all the containers for a given initial stacking configuration. Kim and Bae (1998) present the reshuffle methods for export containers. Kim et al. (2000) derive some decision rules for locating export containers to reduce the number of reshuffles during loading process. Dekker et al. (2006) compare random stacking and category stacking using simulation. It is found that category stacking according to the expected departure time of the containers can lower the number of reshuffles. Wan et al. (2009) propose methods in storage location assignment for export containers that aim at minimizing the number of reshuffles. Zhang et al. (2010) point out the error in Kim et al. (2000) using a counterexample.

* Corresponding author. Tel.: +65 65162895; fax: +65 67771434.
E-mail addresses: xinjia@nus.edu.sg (X. Jiang), iseleelh@nus.edu.sg (L.H. Lee), isecep@nus.edu.sg (E.P. Chew), hanyongbin@nus.edu.sg (Y. Han), kokchoon@nus.edu.sg (K.C. Tan).

0377-2217/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2012.03.011

To balance the high concentration of activities and control the traffic congestion of transporters in the yard, many related studies have been presented. Kim and Kim (1999) study space allocation models for import containers when a segregation strategy is applied in container terminal. Preston and Kozan (2001) develop a model to decide the storage location in order to minimize the turnaround time of all the container vessels. Kim and Kim (2002) present a method to determine the optimal amount of storage space and the optimal number of transfer cranes to handle import containers. Zhang et al. (2003) study the storage allocation problem through a rolling horizon approach. The problem is solved in two stages where in each stage the problem is solved using a mathematical programming approach. Kim and Park (2003) study the allocation of storage space for export containers to utilize space efficiently and to facilitate loading operations. Kozan and Preston (2006) present an iterative search algorithm which integrates a container-transfer model and a container-location model to determine an optimal storage strategy and handling schedule. Vis and Roodbergen (2009) improve the efficiency in the storage yard through scheduling the storage and retrieval of containers. Cordeau et al. (2011) study the car assignment problem in an automotive transshipment terminal to minimize the total handling time.

Both the bad effects of unproductive reshuffles and traffic congestion have been studied particularly for the needs of transshipment hubs in Lee et al. (2006) and Han et al. (2008). For transshipment hubs, loading and unloading activities are usually concentrated and inevitably happen at the same time. This makes the yard planning problem much more challenging compared to port planning for general terminals where loading and unloading activities can be considered independently by having different dedicated storage areas for import and export activities. Lee et al. (2006) propose a consignment strategy to reduce reshuffling to a minimal level by storing export and transshipment containers at dedicated locations according to their destination vessels. A high-low workload balancing protocol is used to reduce the potential traffic congestion of prime movers. A mixed integer programming model is developed to determine the number of incoming containers and the smallest number of yard cranes to deploy in each shift. Their study was extended in Han et al. (2008) to simultaneously consider the space reservation for each vessel.

With the increasing volume of transshipment container handling, the scarcity of storage space is urging new studies to improve the land utilization under the complex requirements of transshipment ports. Although the consignment strategy used is an effective way to reduce reshuffles, the prior reservation of storage spaces for each destination vessel causes under-utilization of land due to the fact that the majority of containers usually come in during the time close to their departure date. In this paper, we propose a space-sharing yard template concept which aims at improving the use of storage space while ensuring the efficiency of yard operations. Storage areas will no longer be reserved entirely for any vessel during the whole planning horizon. Instead, part of the space will be shared by containers staying for different periods.

The rest of this paper is organized as follows. Section 2 provides the detailed description of the problem. Section 3 demonstrates the solution algorithm. Numerical experiments and the computational results are presented in Section 4. Section 5 gives conclusions and some future research topics.

2. Problem description

To provide more flexibility during operation, the terminal we study is divided into sections, and vessels are assigned to sections, each corresponding to several berths, rather than the exact berth locations. Therefore, when we conduct the yard storage allocation within a section, the specific planned berth of a vessel needs not be considered. Furthermore, import containers are not considered in this paper, since they have different characteristics and are usually stored in separate blocks from export and transshipment containers.

To manage the yard allocation process more efficiently, the port operator organizes each section of the storage yard into several blocks as shown in Fig. 1. All different sections of the storage yard are composed of some common basic modules: "sub-block" and "block". To reduce the level of reshuffles, a consignment strategy is used, where export and transshipment containers going to the same destination vessel are stored together in the yard. The smallest unit for the consignment strategy in yard storage allocation process is a "sub-block". The depth of each sub-block is 6 rows of containers, and the length of each sub-block is 8 slots (each slot can accommodate one 20ft container length-wise). The stacking height is 5 containers high (which we call tier). A certain number of sub-blocks in a row form a bigger unit, called a "block". There is a dedicated lane for the movement of prime movers (the "truck path") and a separate "passing lane" strictly to allow trucks to pass each other when required. The passing lane is only wide enough for one prime mover and it is shared between two neighboring container blocks.

According to the workload patterns provided by the port operator, there are two important characteristics of the incoming containers. The characteristics are that the higher incoming workload always happens near to their departure date, while the very low activity happens right after the vessel departs. Hence, it is a common practice for them to use triangular workload profile to do the planning.

For the static yard template (as in Lee et al., 2006; Han et al., 2008), all the sub-blocks in each block have a fixed space capacity, as shown in Fig. 3. This means the maximum amount of space needed at the peak time will be exclusively assigned to each vessel during the whole planning horizon. As much space is only occupied for a short period, it clearly leads to under-utilization of the space. To enjoy the benefit of consignment while increasing the land utilization, we propose a space-sharing method which allows some space to be shared between adjacent neighbors. Essentially it will help to reduce the original space needed for a given workload. As shown in Fig. 3, for the space-sharing yard template, each sub-block has certain amount of storage space for sharing. For example, s12 is the part that can be shared between sub-blocks 1 and 2.

As very few space is needed during the period right after the loading process, the sharing space of one sub-block can be lent to its neighbors. It will then be returned from its neighbors, before the major workload comes into this sub-block. Since the major workload arrives at different periods for different vessels, they will also need the sharing space during different shifts. We can take the sub-block 2 as an example to demonstrate how its space can change over time. Suppose that sub-blocks 1, 2, and 3 have been assigned to different departing vessels, and the starting times of their loading operations are shifts 14, 2 and 5 respectively. Then, the starting times of sharing the spaces to the neighbors for sub-blocks 1, 2, and 3 are shifts 16, 4, and 7 respectively, assuming that the loading operations last for 2 shifts. Since sub-block 2 has sub-blocks 1 and 3 as neighbors, the change of its space capacity over the 21 shifts can be plotted as in Fig. 4. Similarly, the storage space of all the sub-blocks in one block changing over the 21 shifts is shown in Fig. 5. In other words, the space capacity of one sub-block will decrease after the sub-block's loading process, while it increases when its neighbors finish loading. However, the sum of a non-sharing space and its neighboring sharing spaces should be not more than the standard size of a sub-block given by the port operator.

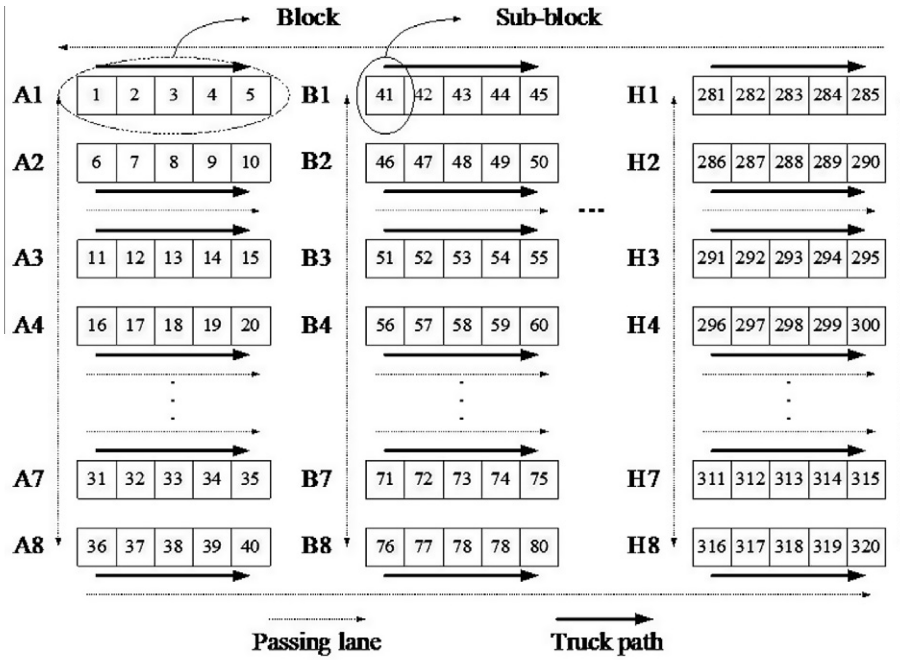

Fig. 1. A storage yard configuration.

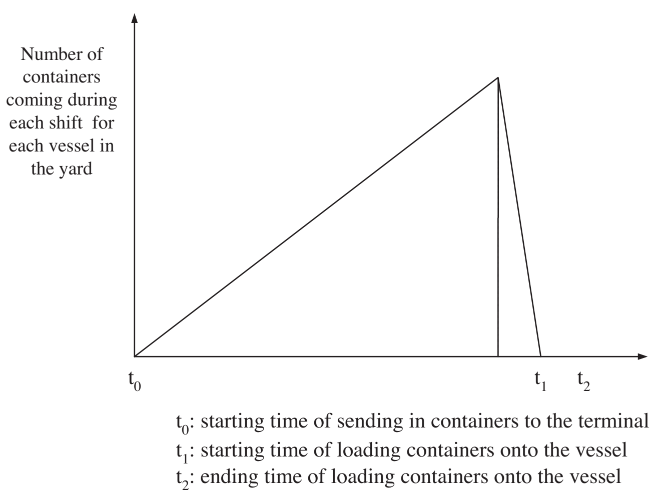

Fig. 2. The buildup pattern of the coming workload for one vessel.

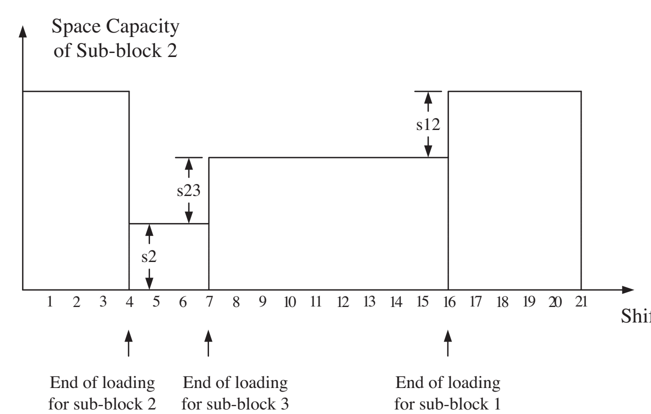

Fig. 4. A schematic diagram for space capacity of one sub-block.

To implement this space-sharing concept, three key issues should be resolved; namely, yard template, size of sharing space and workload assignment.

Since the yard cranes and transporters handle one container at a time, the number of loading and unloading containers in each sub-block can be used to indicate the potential traffic. To ensure a smooth flow of traffic, we adopt the high-low workload balancing protocol and the vicinity matrix from Lee et al. (2006) and Han et al. (2008). The vicinity matrix is used to capture the neighborhood relationship among sub-blocks, while the high-low workload balancing protocol is implemented to avoid potential traffic congestion and to ensure high utilization of yard cranes.

In summary, for this problem, we need to determine a yard template for storage space first. Given the yard template, we will

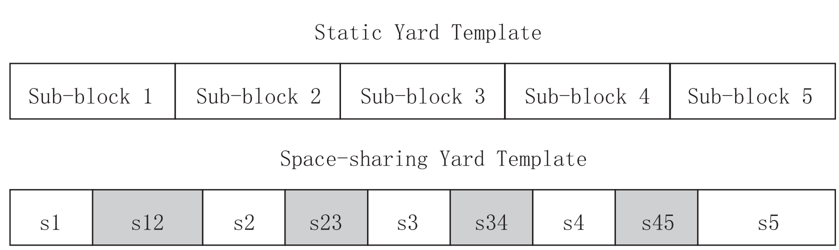

Fig. 3. Schematic diagram of one block for the static yard template and the space-sharing yard template.

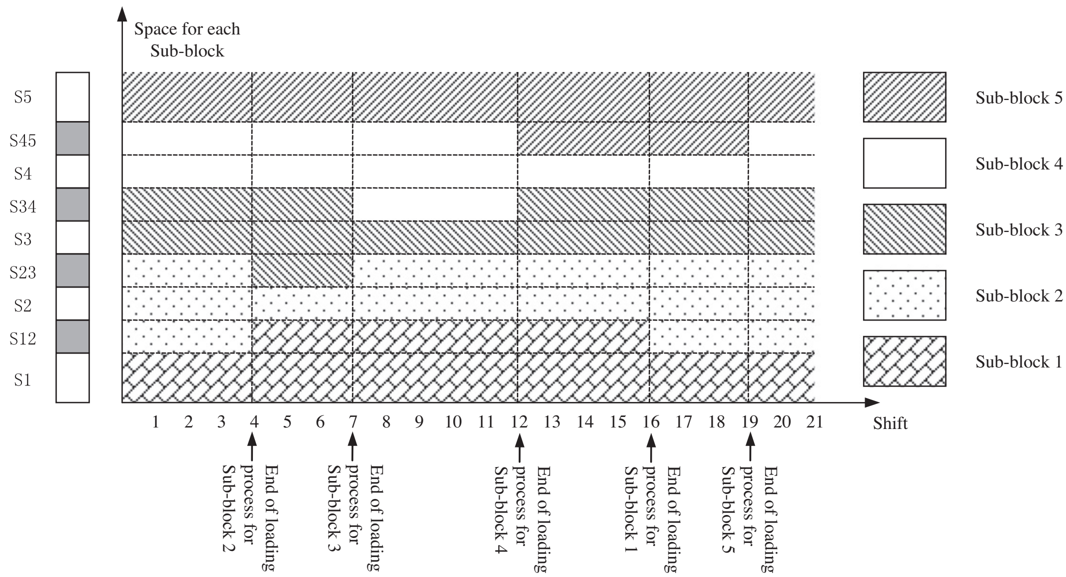

Fig. 5. A schematic diagram for the space capacity of each sub-block in one block.

determine the amount of sharing space between each pair of adjacent sub-blocks to improve the land utilization. Meanwhile the number of containers assigned to each sub-block should also be decided for each shift. According to Han et al. (2008), the port operator does not have any formal planning model to determine the yard template, the sharing space, and container allocation. The decisions are based on yard planners' ingenuity and past experiences. As a means to remedy this, a framework that incorporates the concepts discussed above is developed in the next section.

3. Solution approaches

In this study, there are two main objectives under concern, namely operation efficiency and land productivity. The operation efficiency is measured by the number of yard cranes deployed in a shift. The land productivity measures the amount of space needed in order to handle a certain amount of workload, which is defined as "*total volume of incoming containers/land needed for container handling*". The land productivity can capture the land utilization. As the operation efficiency is more critical in the yard planning, we aims to find the minimum space needed to handle a given amount of workload while ensuring only using a least number of yard cranes in each shift. Since this problem is too complex to solve in an integrated model, a framework which combines space reservation and workload assignment is proposed in this section. The general picture of the framework can be shown in Fig. 6.

The first step is "template generation", which decides the sub-block reservation for each vessel. Based on the information of incoming workload, we can get the requirements of the yard template, such as the minimum number of sub-blocks needed by each vessel. A yard template will then be generated satisfying the requirements.

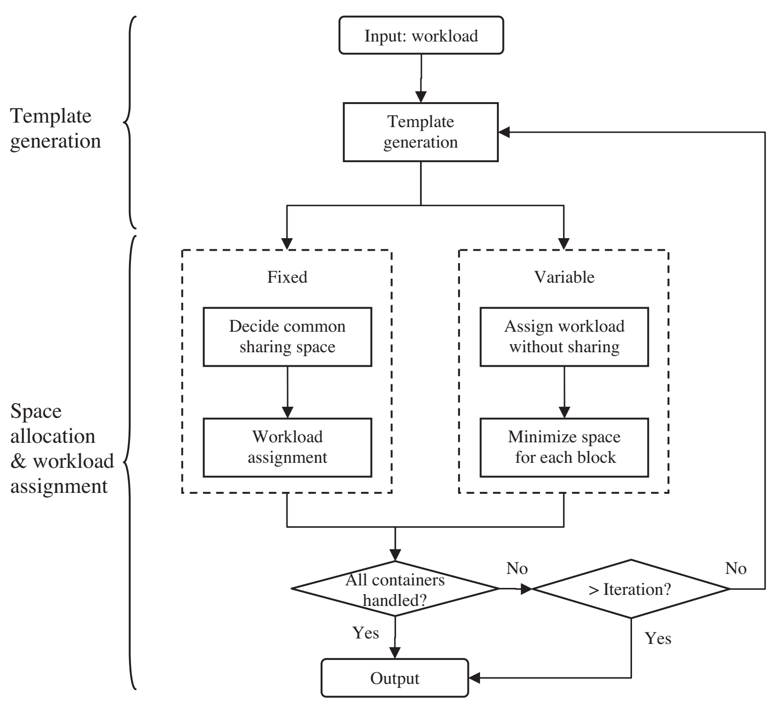

Fig. 6. The framework for solution.

The second step is "space allocation and workload assignment", which decides both the size of sharing space and number of incoming containers for each sub-block in each shift to minimize the space needed. Two different methods can be used to achieve the tasks in this step. The first method is to use a common size of sharing space to simplify the problem. Incoming workload will then be assigned to each sub-block to maximize the common size of sharing space while guarantee the least number of yard cranes deployed in each shift. The details will be presented in the later part of this section. The second method is the variable sharing space method, in which the size of sharing space can be different across sub-blocks. In this method, we first assign the incoming workload without sharing space. Since the space needed in each shift for all sub-blocks are known after the workload assignment, the appropriate size of sharing space between adjacent sub-blocks can be decided through merging the space requests lasting for different periods to minimize the space needed.

To guarantee operation efficiency, the least number of yard cranes from Han et al. (2008) can be adopted to control yard cranes deployed in each shift. If all the containers can be handled within the space-sharing yard template while guarantee the operation efficiency, we get the final solution. Otherwise, the searching will go back to the first step and generate a new yard template as long as the iteration limitation allows.

3.1. Template generation

The yard template is a plan of the yard space reservation for different vessels. It will significantly affect the final result of workload assignment and yard crane deployment. In order to develop a better template, Han et al. (2008) provide an algorithm to generate, repair and improve the template through iteration. In their algorithm, an initial yard template is generated based on the relationship of vessels. Then the template is improved and repaired through swapping sub-blocks reserved for different vessels. The gist of their method is to look into the bottleneck to provide more available space for each vessel in each shift. Some important underlying criteria can be observed from their algorithm:

- In any shift, the number of loading sub-blocks in each block should be no more than one to prevent traffic congestion.
- No neighboring sub-blocks can be loading in the same shift.
- A sub-block is available to receive incoming containers in certain shift only when all its neighbors are not loading in the same shift.
- One sub-block can only be assigned to one vessel (i.e., no mixed stacking).
- The number of sub-blocks assigned to each vessel should be enough to satisfy the space needs of incoming containers.

Since the algorithm is mainly based on pair-wise swapping of sub-blocks, the final template obtained is dependent on the sequence of swapping and the way of searching. To overcome this, a new model is proposed in this section to generate the template directly.

Using the above criteria as constraints, a mathematical programming model can be developed to obtain an effective template. In the proposed model, the bottleneck vessel for the yard template is defined as the vessel that has the smallest number of available sub-blocks in any shift. Similarly, the bottleneck shift is defined as the shift in which the bottleneck vessel has the smallest number of available sub-blocks. In the template generation problem we are trying to get a yard template which provides more available sub-blocks across time. More importantly, the bottleneck vessel and bottleneck shift should be improved to provide more choices for the assignment of incoming workload, which ensures a better solution for the whole space-sharing yard planning problem.

The model parameters are as follows:

A_j the smallest number of sub-blocks that should be reserved for vessel j, $1 \leqslant j \leqslant J$.
B_k the set of sub-blocks that belong to block k, $1 \leqslant k \leqslant K$.
C_k the maximum number of yard cranes allowed to reside in block k at any time, which may vary according to the condition of different blocks, $1 \leqslant k \leqslant K$.
I the number of sub-blocks under consideration.
J the number of vessels under consideration.
K the number of blocks under consideration.
L_{jt} =1, if vessel j is in the loading process in shift t, $1 \leqslant j \leqslant J$, $1 \leqslant t \leqslant T$.
=0, otherwise.
N_i the set of sub-blocks that are neighbors of sub-block i, $1 \leqslant i \leqslant I$.
T the number of shifts under consideration in the planning horizon.
V the lower bound for total number of available sub-blocks in all shifts.

Note: Subscript i is for sub-block, j for vessel, k for block, t for shift.

The decision variables are as follows:

v_{it} =1, if sub-block i is available during shift t, $1 \leqslant i \leqslant I$, $1 \leqslant t \leqslant T$.
=0, otherwise.
z_{ij} =1, if sub-block i is reserved for vessel j, $1 \leqslant i \leqslant I$, $1 \leqslant j \leqslant J$.
=0, otherwise.
u_{ijt} a variable used to indicate the value of $(v_{it} \times z_{ij})$ in linear form.
$1 \leqslant i \leqslant I$, $1 \leqslant i \leqslant I$, $1 \leqslant j \leqslant J$.

The mathematical programming model for the yard template generation problem (denoted as YTG) is as follows:

$$\text{(YTG)} \quad Max\ w \tag{1}$$

$$\text{Subject to}: w \leqslant \sum_{i}^{I} z_{ij} v_{it} \quad \forall 1 \leqslant j \leqslant J, \quad 1 \leqslant t \leqslant T \tag{2}$$

Or equivalently (3)–(6):

$$w \leqslant \sum_{i}^{I} u_{ijt} \quad \forall 1 \leqslant j \leqslant J, \quad 1 \leqslant t \leqslant T \tag{3}$$

$$u_{ijt} \leqslant z_{ij} \quad \forall 1 \leqslant i \leqslant I, 1 \leqslant j \leqslant J, \quad 1 \leqslant t \leqslant T \tag{4}$$

$$u_{ijt} \leqslant v_{it} \quad \forall 1 \leqslant i \leqslant I, \quad 1 \leqslant j \leqslant J, \quad 1 \leqslant t \leqslant T \tag{5}$$

$$u_{ijt} \geqslant 0 \quad \forall 1 \leqslant i \leqslant I, \quad 1 \leqslant j \leqslant J, \quad 1 \leqslant t \leqslant T \tag{6}$$

$$\sum_{j=1}^{J} z_{ij} \leqslant 1 \quad \forall 1 \leqslant i \leqslant I \tag{7}$$

$$\sum_{i=1}^{I} z_{ij} \geqslant A_j \quad \forall 1 \leqslant j \leqslant J \tag{8}$$

$$\sum_{i \in B_k} \sum_{j=1}^{J} z_{ij} L_{jt} < C_k \quad \forall 1 \leqslant k \leqslant K, \quad 1 \leqslant t \leqslant T \tag{9}$$

$$\sum_{i' \in \{i, N_i\}} \sum_{j=1}^{J} z_{i'j} L_{jt} \leqslant 1 \quad \forall 1 \leqslant i \leqslant I, \quad 1 \leqslant t \leqslant T \tag{10}$$

$$1 - v_{it} \leqslant \sum_{i' \in N_i} \sum_{j=1}^{J} z_{i'j} L_{jt} \leqslant 3 - 3 v_{it} \quad \forall 1 \leqslant i \leqslant I, \quad 1 \leqslant t \leqslant T \tag{11}$$

$$\sum_{t}^{T} \sum_{i}^{I} v_{it} \geqslant V \tag{12}$$

$$v_{it} \in \{0, 1\} \quad \forall 1 \leqslant i \leqslant I, \quad 1 \leqslant t \leqslant T \tag{13}$$

$$z_{ij} \in \{0, 1\} \quad \forall 1 \leqslant i \leqslant I, \quad 1 \leqslant j \leqslant J \tag{14}$$

The objective is to maximize the value of the bottleneck. We use w to represent the value, therefore for any vessel in any shift, the number of available sub-blocks should be bigger than w as shown in Constraint (2). Since both v_{it} and z_{ij} are 0–1 integers, Constraint (2) is equal to the linear constraints (3)–(6). Constraint (7) ensures that a sub-block can be reserved for at most one vessel during the whole planning horizon and no change in the reservation can be made once the reservation is made. However, part of the sub-block can be shared with its neighbors. Constraint (8) ensures that sufficient sub-blocks are reserved for each vessel. Constraint (9) ensures that the number of sub-blocks in loading process should be less than the maximum number of yard cranes assigned to each block in each shift; otherwise the rest sub-blocks in that block will be unavailable during the shift. Constraint (10) ensures that no neighboring sub-blocks should be loading in the same shift to avoid potential traffic congestion. Constraint (11) defines the availability of a sub-block. According to the configuration of the template, a sub-block can have 3 neighbors at most. Therefore, when the number of loading neighbors is 1–3, the sub-block is unavailable; when the number of loading neighbors is 0, the sub-block is available. Constraint (12) ensures that the number of available sub-blocks should be bigger than the requirement. Constraints (13) and (14) are 0 or 1 value restrictions for decision variables.

The YTG model is difficult to solve to optimality directly using CPLEX. Fortunately we do not need to run the model to optimality since the purpose of this model is to get a good initial template to be used for subsequent steps. In our implementation, we will set required values for the objective and V in constraint (12) to provide feasible solutions efficiently. The yard template will then be tested in "space allocation & workload assignment" to see if it is good enough. If the incoming containers cannot be handled in step 2, the required values in step 1 will increase accordingly to get a new yard template for the next iteration.

Later in our numerical run, we will show that the YTG model performs better than the algorithm in Han et al. (2008) because it can generate feasible template for the higher workload level.

3.2. Two different methods for space allocation and workload assignment

After the template generation step, the yard template will be used for space allocation and workload assignment. In this step, both the size of the sharing space and workload assignment should be decided. Two different methods are presented in this section to achieve these two tasks.

3.2.1. Fixed sharing space

In the "fixed sharing space method", a common size of sharing space is used across the yard template to simply the problem. When the common size is used, each sub-block will be divided into non-sharing space and sharing space with neighboring sub-blocks. As the size of sub-block is fixed and the number of sub-blocks in a block is constant, the larger scale of common size means more sharing space and so the overall space in a block is reduced, which can be shown in Fig. 7. In order to improve the land productivity, we need to maximize the common size of sharing space to handle all containers.

The model parameters are as follows:

- CC the capacity of each yard crane in terms of container moves per shift, which is 100 in this model.
- CLS_i the left space capacity of non-sharing space for sub-block i, $1 \leqslant i \leqslant I$.
- $CLS_{ii'}$ the left space capacity of sharing space between sub-block i and i', $1 \leqslant i \leqslant I$, $1 \leqslant i' \leqslant I$.
- CLC_i the left yard crane capacity of sub-block i for loading process, $1 \leqslant i \leqslant I$.
- CS the original space capacity of each sub-block in terms of TEUs, which is 240 (5 tiers × 6 lanes × 8 slots) in this model.
- HL the lowest value that a high workload can take.
- HU the highest value that a high workload can take.
- LB_t the least number of yard cranes can be deployed for unloading in shift t, $1 \leqslant t \leqslant T$.
- LL the lowest value that a low workload can take.
- LU the highest value that a low workload can take.
- M a sufficiently large positive value.
- NB_i the set of adjacent sub-blocks of sub-block i, where space sharing is possible between sub-block i and i', $1 \leqslant i \leqslant I$.
- S_{ti} the set of shifts from the end of the loading time of sub-block i to the current shift t, $1 \leqslant i \leqslant I$, $1 \leqslant t \leqslant T$.
- $W_{ii't}$ =1, if the sharing space between sub-blocks i and i' ($s_{ii'}$) belongs to part of capacity of sub-block i in shift t. $W_{ii't}$ can be obtained from the loading time of sub-blocks i and i', $1 \leqslant i \leqslant I$, $1 \leqslant i' \leqslant I$, $1 \leqslant t \leqslant T$.
  =0, if the sharing space between Sub-blocks i and i' ($s_{ii'}$) belongs to part of capacity of sub-block i' in Shift t.
- WX_{jt} the number of 20ft containers arriving at the terminal in shift t and will be loaded onto vessel j finally. It is given and input to the model, $1 \leqslant j \leqslant J$, $1 \leqslant t \leqslant T$.
- WY_{jt} the number of 40ft containers arriving at the terminal in shift t and will be loaded onto vessel j finally. It is given and input to the model, $1 \leqslant j \leqslant J$, $1 \leqslant t \leqslant T$.
- Z_{ij} $=z_{ij}$, indicator of sub-block reservation for different vessels, adopting the value solved in template generation step, $1 \leqslant i \leqslant I$, $1 \leqslant j \leqslant J$.

Note: Subscript i is for sub-block, j for vessel, k for block, t for shift.

The decision variables are as follows:

- d_{kt} the number of yard cranes allocated to block k for unloading in shift t, $1 \leqslant k \leqslant K$, $1 \leqslant t \leqslant T$.
- h_{it} =1, if the total workload allocated to sub-block i for unloading in shift t is high, that is, $HL \leqslant x_{it} + y_{it} \leqslant HU$, $1 \leqslant i \leqslant I$, $1 \leqslant t \leqslant T$.
  =0, if the total workload allocated to sub-block i for unloading in shift t is low, that is, $LL \leqslant x_{it} + y_{it} \leqslant LU$, $1 \leqslant i \leqslant I$, $1 \leqslant t \leqslant T$.
- s the common size of sharing space between each pair of adjacent sub-blocks.
- s_i the space that belongs to sub-block i and cannot shared with its neighbors, $1 \leqslant i \leqslant I$.
- $s_{ii'}$ the space that can be shared between sub-blocks i and i',

(*continued on next page*)

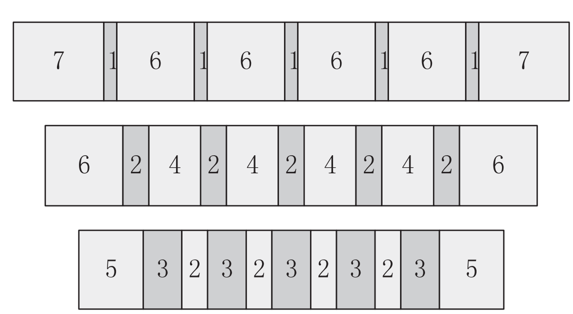

Fig. 7. Block configurations with common size = 1, 2, 3 slots respectively.

$1 \leqslant i \leqslant I, 1 \leqslant i' \leqslant I.$

x_{it} the number of 20ft containers that are allocated to sub-block i for unloading in shift t, $1 \leqslant i \leqslant I, 1 \leqslant j \leqslant J, 1 \leqslant t \leqslant T$.

y_{it} the number of 40ft containers that are allocated to sub-block i for unloading in shift t, $1 \leqslant i \leqslant I, 1 \leqslant j \leqslant J, 1 \leqslant t \leqslant T$.

The workload assignment problem (WAP) can be developed as follows to maximize the common size of sharing space.

$$\text{(WAP)}\ \ Max\, s \tag{15}$$

$$\text{Subject to}: \sum_{i=1}^{I} Z_{ij} x_{it} = WX_{jt} \quad \forall 1 \leqslant j \leqslant J,\ 1 \leqslant t \leqslant T \tag{16}$$

$$\sum_{i=1}^{I} Z_{ij} y_{it} = WY_{jt} \quad \forall 1 \leqslant j \leqslant J,\ 1 \leqslant t \leqslant T \tag{17}$$

$$s_{ii'} = 30 \times s \quad \forall 1 \leqslant i \leqslant I,\ 1 \leqslant i' \leqslant I \tag{18}$$

$$s_i + \sum_{i' \in NB_i} s_{ii'} = CS \quad \forall 1 \leqslant i \leqslant I \tag{19}$$

$$\sum_{t' \in S_{it}} (x_{it} + 2y_{it}) \leqslant s_i + \sum_{i' \in NB_i} W_{ii't} s_{ii'} \quad \forall 1 \leqslant i \leqslant I,\ 1 \leqslant t \leqslant T \tag{20}$$

$$\sum_t (x_{it} + y_{it}) \leqslant \sum_j \left(Z_{ij} \sum_t L_{jt} \times CC \right) \quad \forall 1 \leqslant i \leqslant I \tag{21}$$

$$\sum_{i \in B_k} (x_{it} + y_{it}) \leqslant d_{kt} CC \quad \forall 1 \leqslant k \leqslant K \tag{22}$$

$$d_{kt} + \sum_{i \in B_k} \sum_{j=1}^{J} Z_{ij} L_{jt} \leqslant C_k \quad \forall 1 \leqslant k \leqslant K,\ 1 \leqslant t \leqslant T \tag{23}$$

$$\sum_k d_{kt} = LB_t \quad \forall 1 \leqslant t \leqslant T \tag{24}$$

$$HL + (LL - HL)(1 - h_{it}) \leqslant x_{it} + y_{it} \leqslant LU + (HU - LU) h_{it} \quad \forall 1 \leqslant i \leqslant I,\ 1 \leqslant t \leqslant T \tag{25}$$

$$\sum_{i' \in N_i} (x_{i't} + y_{i't}) \leqslant M \left(1 - \sum_{j=1}^{J} Z_{ij} L_{jt} \right) \quad \forall 1 \leqslant i \leqslant I,\ 1 \leqslant t \leqslant T \tag{26}$$

$$\sum_{i' \in N_i \cup \{i\}} h_{i't} \leqslant 1 \quad \forall 1 \leqslant i \leqslant I,\ 1 \leqslant t \leqslant T \tag{27}$$

$$s \in \{0, 1, 2, 3, 4\} \tag{28}$$

$$x_{it} \geqslant 0 \ \ y_{it} \geqslant 0 \quad \forall 1 \leqslant i \leqslant I,\ 1 \leqslant t \leqslant T \tag{29}$$

$$h_{it} \in \{0, 1\} \quad \forall 1 \leqslant i \leqslant I,\ 1 \leqslant t \leqslant T \tag{30}$$

$$d_{kt} \in \{Positive\ Integer\} \quad \forall 1 \leqslant k \leqslant K, 1 \leqslant t \leqslant T \tag{31}$$

Constraints (16) and (17) ensure that all the workload arriving at the terminal in each shift for each vessel will be allocated to corresponding storage locations. Constraints (18) and (19) are used to calculate the size of sharing and non-sharing space for each sub-block. Constraint (20) ensures the space capacity restriction of each sub-block during each shift. The capacity includes the fixed part and the sharing part with its neighbors. Constraint (21) ensures that the containers in each sub-block should be loaded onto the destination vessel within a certain time span. Constraint (22) ensures that the yard cranes allocated to each block for unloading can handle all the unloading workload in each shift. As a result of the limitation of the length of the chassis trailer and due to safety consideration, each block can hold at most a certain number of yard cranes at any one time. Constraint (23) ensures this restriction. In addition, one yard crane is required for each sub-block in the loading process, and hence the number of sub-blocks in the loading process is exactly equal to the number of yard cranes assigned to that block for loading. Constraint (24) guarantees the least number of yard cranes deployed in each shift. To make full use of yard cranes, workload allocated to each sub-block in each shift should be either high or low. In this model, constraint (25) is used to ensure this restriction. Constraint (26) ensures that all the neighbors of a sub-block in the loading process cannot accept any workload in that shift. Constraint (27) ensures that high unloading workload cannot be allocated to two sub-blocks that are neighbors of each other in the same shift. Constraints (28)–(31) are non-negative and integer restrictions.

As the maximum size of a sub-block is 8-slot, the scale of common size of sharing space can only take the integer value from 0 to 4 slots, as shown in constraint (28). In this case, we can set the objective value to different scale of common size and solve WAP for feasible solution, while the solution for the largest scale of common size is kept as optimal. Besides, the only connection constraints from shift to shift are (20) and (21). If we replace these two constraints with (32) and (33) using remaining space capacity and loading capacity in shift t, the workload assignment with fixed sharing space can be solved efficiently using the sequential method in Lee et al. (2006).

$$x_{it} + 2y_{it} \leqslant CLS_i + \sum_{i' \in NB_i} W_{ii't} CLS_{ii'} \quad \forall 1 \leqslant i \leqslant I \tag{32}$$

$$x_{it} + y_{it} \leqslant CLC_i \quad \forall 1 \leqslant\ i \leqslant I \tag{33}$$

3.2.2. Variable sharing space

In the previous method, the same size of sharing space is used for the yard template. Since the workload assignment does not match the same size, there will always be gaps of unused space. To fit variable size of sharing space to incoming containers, we can first assign the workloads as in static yard template planning. Then the size of sharing and non-sharing space can be decided based on the results of workload assignment.

The workload assignment based on the static yard template can be achieved using the WAP model by setting common size of sharing space s as 0-slot. Based on the workload assignment solution of x_{it} and y_{it}, we now know the incoming and retrieval of containers for each sub-block. Then, the slots needed by each sub-block in each shift can be calculated with the remaining containers in the sub-block. If we number all the slots in a block as a sequence, the slots occupied by containers in two example sub-blocks can be shown in Fig. 8. As the size of each sub-block is assumed to be 8 slots when we solve the workload assignment problem, sub-block 1 will use the slots chosen from slots 1 to 8, while sub-block 2 choosing from slots 9 to 16. To describe the problem more clearly, we use range $\left(S^L_{it}, S^U_{it}\right)$ to represent that the slot $S^L_{it} + 1$ to slot S^U_{it} are occupied by sub-block i in shift t. For instance, (8, 16) is used by sub-block 2 in shift 4, which means slots 9–16 are occupied.

Once a slot has incoming containers, it will be occupied until the end of loading process of the sub-block. Hence if a slot is occupied by a sub-block in the current shift, the slot should also be assigned to the same sub-block in the subsequent shift until the loading process ends. It can be found in Fig. 8 that, slots 13–16 are not occupied by sub-block 2 from shift 5 to 14, while slots 5–8 are occupied by sub-block 1 only from shift 7 to 14. Therefore, 4 slots can actually be shared by the two sub-blocks, as they are occupied during different shifts. The size of sharing space can be decided between these two adjacent sub-blocks, as shown in Fig. 9.

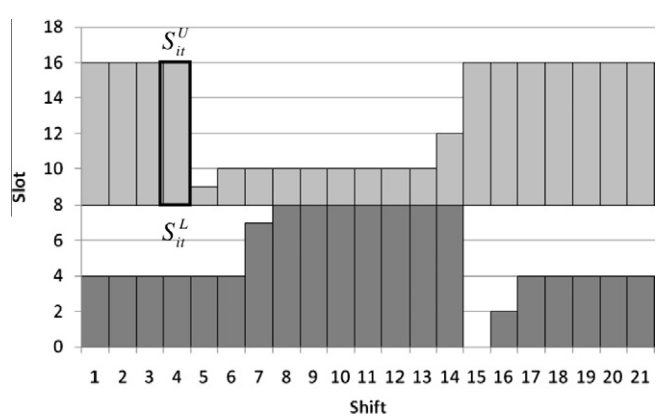

Fig. 8. Relative locations occupied by two sub-blocks without sharing.

Since space can only be shared between adjacent neighbors, to ensure no reshuffling, the size of each sharing space can be decided block by block. Based on the idea of variable sharing space method, we can develop the following model to solve the space-sharing problem. In this model, the objective is to minimize the total number of slots occupied by the whole block. The model parameters are as follows:

F_i The shift when sub-block i finishes its loading process.
sub The number of sub-blocks in each block.

The decision variables are as follows:

S_{it}^L The lower boundary of the slot range occupied by sub-block i in shift t.
S_{it}^U The higher boundary of the slot range occupied by sub-block i in shift t.
S_{it}^{40} The pair of adjacent of slots needed for 40ft containers needed by sub-block i in shift t.

The minimal block space problem (MBS) can be developed as follows:

$$(\text{MBS})\ \ Min\, w \tag{34}$$

$$\text{Subject to}: w \geqslant S_{sub,t}^U \ \ \forall 1 \leqslant t \leqslant T \tag{35}$$

$$\left(S_{it}^U - S_{it}^L\right) \times 30 \geqslant \sum_{t=F_i+1}^{t} (x_{it} + 2 \times y_{it}) \ \ \forall 1 \leqslant i \leqslant sub,\ 1 \leqslant t \leqslant T \tag{36}$$

$$S_{it}^{40} \geqslant \frac{1}{2} \times \sum_{t=F_i+1}^{t-1} x_{it} + \sum_{t=F_i+1}^{t} y_{it} \ \ \forall 1 \leqslant i \leqslant sub,\ 1 \leqslant t \leqslant T \tag{37}$$

$$S_{it}^U - S_{it}^L \geqslant 2 \times S_{it}^{40} \ \ \forall 1 \leqslant i \leqslant sub,\ 1 \leqslant t \leqslant T \tag{38}$$

$$S_{it}^U - S_{it-1}^U \geqslant 0 \ \ \forall 1 \leqslant i \leqslant sub,\ 1 \leqslant t \leqslant T,\ t \neq F_i + 1 \tag{39}$$

$$S_{it}^L - S_{it-1}^L \leqslant 0 \ \ \forall 1 \leqslant i \leqslant sub,\ 1 \leqslant t \leqslant T,\ t \neq F_i + 1 \tag{40}$$

$$S_{(i+1)t}^L - S_{it}^U \geqslant 0 \ \ \forall 1 \leqslant i < sub,\ 1 \leqslant t \leqslant T \tag{41}$$

$$S_{it}^U - S_{it}^L \leqslant 8 \ \ \forall 1 \leqslant i < sub,\ 1 \leqslant t \leqslant T \tag{42}$$

$$S_{it}^U \geqslant 0 \ \ S_{it}^L \geqslant 0 \ \ \forall 1 \leqslant i \leqslant sub,\ 1 \leqslant t \leqslant T \tag{43}$$

$$S_{it}^L, S_{it}^U \in \{Positive\, Integer\} \ \ \forall 1 \leqslant i \leqslant sub,\ 1 \leqslant t \leqslant T \tag{44}$$

Constraint (35) captures the maximum slot occupation by the whole block. Constraint (36) ensures that the slots occupied sub-block i during shift t is larger than the total space needed by containers stored in the sub-block. In our study, the 40ft containers need two consecutive slots for storage, and space reserve for them can also store 20ft containers. Therefore we use constraints Eq. (37) to make sure that the incoming 40ft containers will always have space to store without reshuffles. Constraint (38) is then used to ensure that the pairs of slots needed by 40ft containers are included in the range occupied by the corresponding sub-block. Constraints (39) and (40) ensure that the space occupied by a sub-block at shift t will always be covered by the same sub-block at the following shift before the completion of loading procedure. Constraint (41) ensures that a space unit cannot be occupied by different sub-blocks at the same time. The maximum number of slots allowed in a sub-block is guaranteed in Constraint (42). Constraints (43) and (44) are non-negative and integer restrictions.

The MBS model can be solved directly in a short time. The size of sharing and non-sharing space can be calculated with the following two equations. The size of sharing space between sub-block i and $i+1$ is

$$S_{i(i+1)} = \max_t S_{it}^U - \min_t S_{(i+1)t}^L \tag{45}$$

The size of non-sharing space for sub-block i is

$$S_i = \min_t S_{(i+1)t}^L - \max_t S_{(i-1)t}^U \tag{46}$$

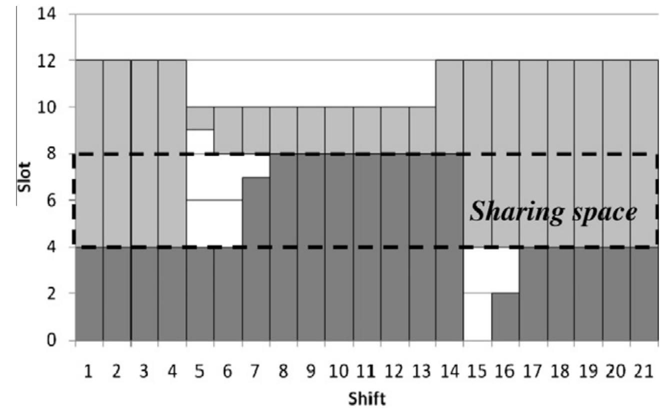

Fig. 9. Relative locations occupied by two sub-blocks with sharing.

4. Numerical experiments

In this section, the solution procedure is implemented in C++ and run on the same computer as that for the static yard template problem. In all the numerical examples, the general scale of the problem is 64-block, 21-vessel and 21-shift.

4.1. Experiment descriptions

In order to test the robustness of our results and algorithms, we have generated different realizations of the workload based on the triangular pattern shown in Fig. 2, and we have also varied the overall workload per week to test out different port throughput levels.

The "even pattern" and "wave pattern" are used to test two different situations of the workload, as shown in Fig. 10. For there "even pattern", there are no major variations in the workload among different shifts. For the "wave pattern", the workload is highly uneven, i.e., in some shifts, the workload is relative heavy; while in other shifts, the workload is relatively light. The "wave pattern" can be seen as the worst case scenario. Meanwhile, the workload for each vessel still follows the triangular form.

For both "even pattern" and "wave pattern" scenarios, we first vary the total workload that is coming to the port. The workload level is the total volume of containers coming during the whole planning horizon (one week), which varies from 46,000 TEU/week to 80,000 TEU/week (50–85% of the original space given). Then for each workload level, we randomly generate ten different scenarios. Even through the general workload of a vessel still follows the triangular distribution, the parameters for each vessel may vary because of uncertainty. For example, the peak of containers arrival can vary from 3 to 10 shifts before the loading procedure. In addition, the volume of incoming containers in each shift may vary ±10% from the parameter set by the triangular distribution. In all, we have run more than 700 randomly generated scenarios.

4.2. Experiment results

The performance of "fixed sharing space" and "variable sharing space" for space-sharing yard template planning is presented in Figs. 11 and 12 respectively. The land productivity achieved by the static yard template (which is without sharing) is also included in the figures and by definition, if we can find a feasible container allocation for a given workload level, the land productivity will be just equal to the total incoming workload divided by the original space given. For the space-sharing yard template planning, we will use the "fixed sharing space" and "variable sharing space" methods to find the sharing space, which eventually reduces the original

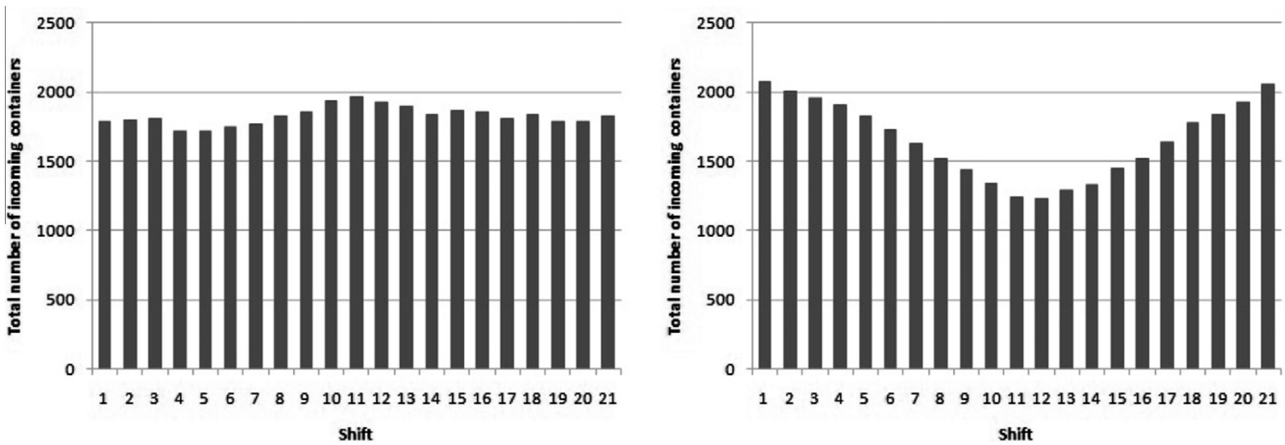

Fig. 10. Input data with "even pattern" (left) and "wave pattern" (right).

space given, and so the land productivity increases in these two cases.

Figs. 11 and 12 show the land productivity at different workload levels. For each workload level, we generate 10 sets of random input data. The highest, the middle and the lowest points for each workload level represent respectively the maximum, the average and the minimum values based on the 10 sets of run. For most of the cases, the running time is within minutes, but for some worst cases, the running time can be hours and this is mainly caused by the high-low workload constraints.

Comparing the land productivity achieved by space-sharing yard template and static yard template under different workload levels, the following results can be observed.

Both the proposed methods can improve the land productivity when compare with the static model (Han et al., 2008). Note that for the consignment strategy, the static yard template planning can never achieve a land productivity exceeding a value of 1. However, the space-sharing yard template can achieve land productivity greater than 1 because each unit of storage space can accommodate up to 2 different types of containers while satisfying the constraints of consignment.

Among the two methods, the variable sharing space method performs better. Firstly, its land productivity dominates the other. Secondly, it has lower variability. The reason is that, the "variable sharing space method" is able to have different sizes of sharing space for different sub-block, while the "fixed sharing space method" does not have that flexibility.

Moreover, the "variable sharing space method" is more robust. We conduct statistical test to show if the land productivity is affected by the different patterns of workload. The results indicate that for the "fixed sharing space method", we can achieve higher land productivity for the "even pattern" workload compare with the "wave pattern" workload. On the other hand, for the "variable sharing space method", we cannot show that there is a significant difference in the land productivity between the two patterns, and this shows the robustness of the approach. The results for the fixed sharing space are not surprising. This is because for the wave pattern, the workload for the shifts is highly uneven, and therefore the space has to be catered for the worst case. Fixing the common size of the sharing space across the blocks will limit its flexibility in handling this uneven workload. On the other hand, for the variable sharing space, it has the flexibility to adjust, and for those with high workload, the sharing space is less, but for those with low workload, the sharing space can be more.

On the other hand, the YTG model proposed in this paper performs better than the algorithm in Han et al. (2008). In the term of the computation time, it is comparable to the Han's algorithm but in term of the solution quality, it is better. In our numerical example for the static yard template planning problem, Han's algorithm cannot provide feasible solutions at the higher workload

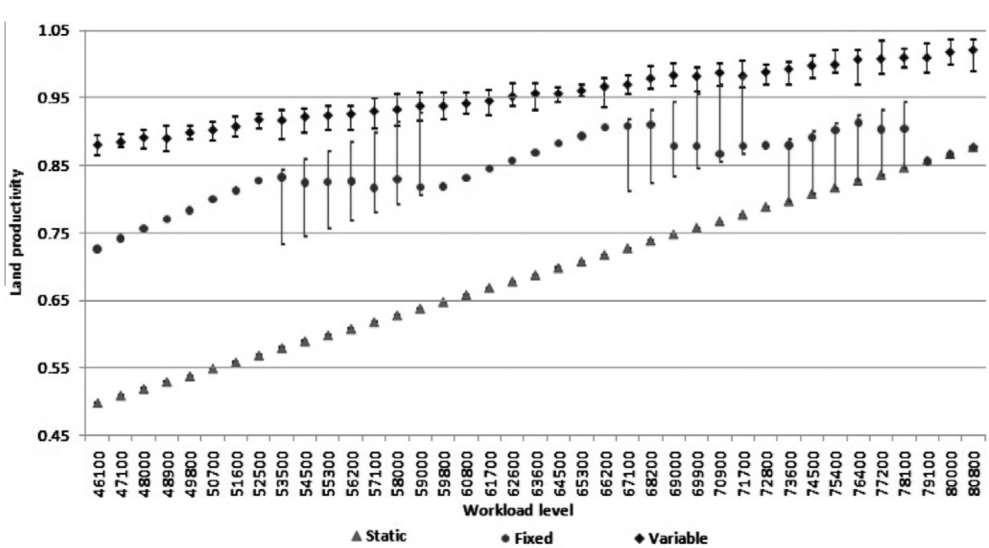

Fig. 11. Results from "even pattern" of input data.

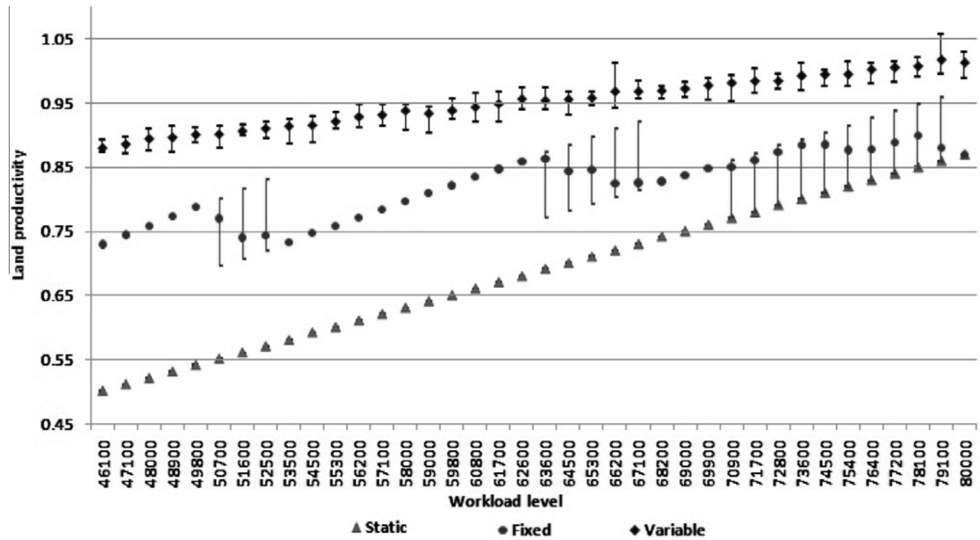

Fig. 12. Results from "wave pattern" of input data.

level. For example at the workload level of 80,800 TEU/week for the "even pattern" and 80,000 TEU/week for the "wave pattern", Han's algorithm cannot provide a feasible template while YTG model is able to provide a feasible solution.

5. Conclusions and future research

In this paper we study an actual problem faced by a leading transshipment port operator. Currently, the port operator uses a consignment strategy to reduce the reshuffling level for high equipment productivity. However, this strategy leads to under-utilization of storage space, due to pre-reservation of the maximum space needed during the whole planning horizon. With the increasing volume of transshipment container handling, the scarcity of storage space is urging new studies to balance the equipment productivity and the land usage. A new approach named space-sharing yard template planning, is studied in this paper to improve the land utilization when the consignment strategy is implemented.

We develop a framework which integrates space reservation and workload assignment to solve the space-sharing yard template problem. In this framework, two different approaches are proposed to determine the sharing space namely, "fixed sharing space method" and "variable sharing space method". The numerical experiments show that, both the proposed methods can improve the land productivity under different port throughput levels compared to the static yard planning method which do not allow the sharing of space. The space-sharing yard template planning can even achieve a land productivity exceeding a value of 1, which is impossible for a template without sharing. Among the two methods, the variable sharing space method performs better both in land productivity and robustness.

However, the study has its own limitation. In this work, the problem is considered at the planning level. As the size of sharing and non-sharing space is fixed during planning, some recovery strategies should be developed to handle the uncertainties in operational level. Future research directions can consider uncertainties, and possibly integrate the planning model with the real time operation.

References

Cordeau, J.-F., Laporte, G., Moccia, L., Sorrentino, G., 2011. Optimizing yard assignment in an automotive transshipment terminal. European Journal of Operational Research 215, 149–160.

De Castilho, B., Daganzo, C.F., 1993. Handling strategies for import containers at marine terminals. Transportation Research Part B 27, 151–166.

Dekker, R., Voogd, P., Asperen, E., 2006. Advanced methods for container stacking. OR Spectrum 28, 563–586.

Han, Y., Lee, L.H., Chew, E.P., Tan, K.C., 2008. A yard storage strategy for minimizing traffic congestion in a marine container transshipment hub. OR Spectrum 30, 697–720.

Kim, K.H., 1997. Evaluation of the number of reshuffles in storage yards. Computers and Industrial Engineering 32, 701–711.

Kim, K.H., Bae, J.W., 1998. Re-marshaling export containers in port container terminals. Computers and Industrial Engineering 35, 655–658.

Kim, K.H., Kim, H.B., 1999. Segregating space allocation models for container inventories in port container terminals. International Journal of Production Economics 59, 415–423.

Kim, K.H., Park, Y.M., Ryu, K.-R., 2000. Deriving decision rules to locate export containers in container yards. European Journal of Operational Research 124, 89–101.

Kim, K.H., Kim, H.B., 2002. The optimal sizing of the storage space and handling facilities for import containers. Transportation Research Part B 36, 821–835.

Kim, K.H., Park, K.T., 2003. A note on a dynamic space-allocation method for outbound containers. European Journal of Operational Research 148, 92–101.

Kozan, E., Preston, P., 2006. Mathematical modeling of container transfers and storage locations at seaport terminals. OR Spectrum 28, 519–537.

Ku, L.P., Lee, L.H., Chew, E.P., Tan, K.C., 2010. An optimization framework for yard planning in a container terminal: case with automated rail-mounted gantry cranes. OR Spectrum 32, 519–541.

Lee, L.H., Chew, E.P., Tan, K.C., Han, Y., 2006. An optimization model for storage yard management in transshipment hubs. OR Spectrum 28, 539–561.

Preston, P., Kozan, E., 2001. An approach to determine storage locations of containers at seaport terminals. Computers & Operations Research 28, 983–995.

Steenken, D., Voβ, S., Stahlbock, R., 2004. Container terminal operation and operations research – a classification and literature review. OR Spectrum 26, 3–49.

Stahlbock, R., Voβ, S., 2008. Operations research at container terminals: a literature update. OR Spectrum 20, 1–52.

Vis, I.F.A., de Koster, R., 2003. Transshipment of containers at a container terminal: an overview. European Journal of Operational Research 147, 1–16.

Vis, I.F.A., Roodbergen, K.J., 2009. Scheduling of container storage and retrieval. Operations Research 57, 456–467.

Wan, Y., Liu, J., Tsai, P.C., 2009. The assignment of storage locations to containers for a container stack. Naval Research Logistics 56, 699–713.

Zhang, C., Liu, J., Wan, Y.W., Murty, K.G., Linn, R.J., 2003. Storage space allocation in container terminals. Transportation Research Part B 37, 883–903.

Zhang, C., Chen, W., Shi, L., Zheng, L., 2010. A note on deriving decision rules to locate export containers in container yards. European Journal of Operational Research 205, 483–485.