

Loo Hay Lee · Ek Peng Chew · Kok Choon Tan · Yongbin Han

An optimization model for storage yard management in transshipment hubs

Published online: 28 April 2006
© Springer-Verlag 2006

Abstract This paper studies a yard storage allocation problem in a transshipment hub where there is a great number of loading and unloading activities. The primary challenge is to efficiently shift containers between the vessels and the storage area so that reshuffling and traffic congestion is minimized. In particular, to reduce reshuffling, a consignment strategy is used. This strategy groups unloaded containers according to their destination vessel. To reduce traffic congestion, a new workload balancing protocol is proposed. A mixed integer-programming model is then formulated to determine the minimum number of yard cranes to deploy and the location where unloaded containers should be stored. The model is solved using CPLEX. Due to the size and complexity of this model two heuristics are also developed. The first is a sequential method while the second is a column generation method. A bound is developed that allows the quality of the solution to be judged. Lastly, a numerical investigation is provided and demonstrates that the algorithms perform adequately on most cases considered.

Keywords Port operation · Storage allocation · Mixed-integer programming · Heuristic algorithm · Column generation

1 Introduction

Container traffic has been growing steadily and this trend is expected to continue (Yun and Choi 1999; Ryan 1998). A new generation of container vessels that have a greater carrying capacity and scarcity of the land will put an enormous pressure on port operators to develop effective container handling systems. High-density, automated container handling equipment is a potential candidate for improving the performance of container terminals and meeting the challenges of the future in

L. H. Lee · E. P. Chew · K. C. Tan · Y. Han (✉)

Department of Industrial and Systems Engineering, National University of Singapore,
10 Kent Ridge, 119260 Singapore, Singapore

E-mail: iselee@nus.edu.sg, isecep@nus.edu.sg, isetkc@nus.edu.sg, hanyongbin@nus.edu.sg

marine transportation. However, in order for these capital intensive equipments to function effectively, decision planning tool for integration and optimization becomes crucial.

A container terminal is the place where vessels dock on a berth and containers are loaded and unloaded. Based on the types of container handling operations, a container terminal can be roughly divided into two main areas, the quayside for berthing vessels and the storage yard for holding containers (as shown in Fig. 1). The quayside is made up of several berths for vessels to moor. The vessels moored at the berths are served by quay cranes (QCs) which load and unload containers. The storage yard is typically divided into several blocks where the containers are stored. Each container block is served by several yard cranes (YCs). The storage yard serves as an interface for loading (unloading) containers to (from) vessels to facilitate export (import) containers and to transship containers between vessels. To transport containers between the quayside and the storage yard, vehicles such as prime mover, or straddle carriers are used. A schematic diagram of the typical

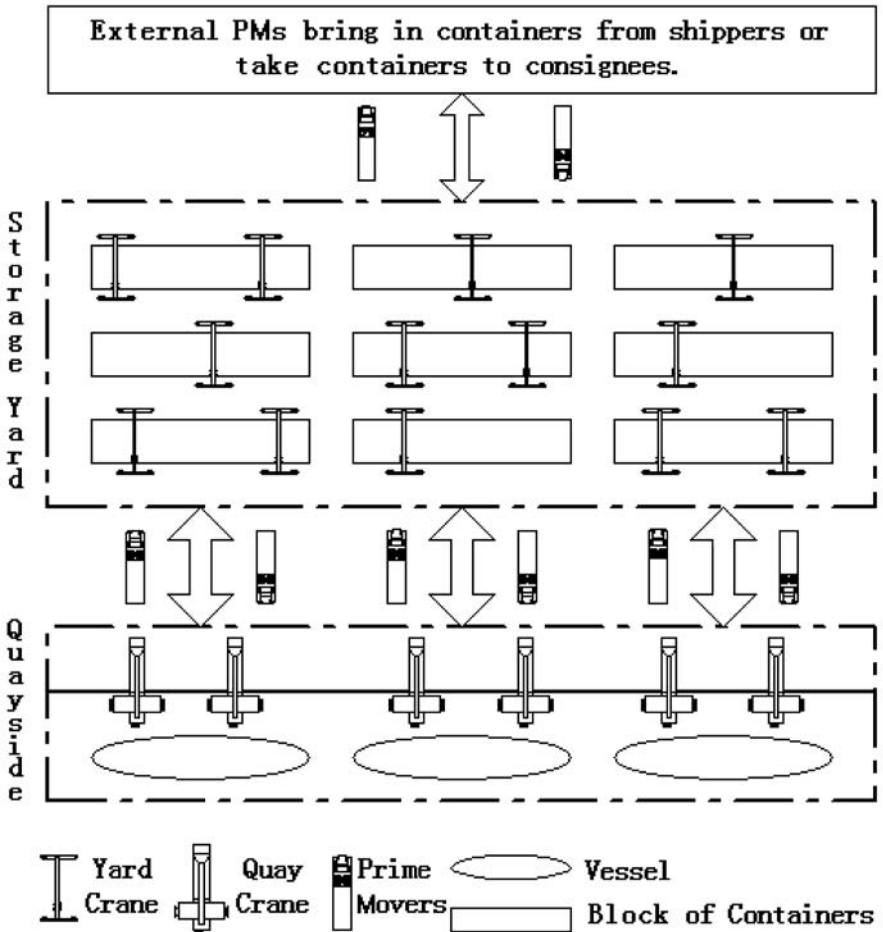


Fig. 1 A schematic diagram of a container terminal

processes in a container terminal is shown in Fig. 2 (Vis and De Koster 2003). The container activities can be categorized into three types: import, export, and transshipment activities. For export activities, the containers are brought in by shippers and will be stored at their designated locations in the storage yard. When it is time to load the containers, they are retrieved from the stored location and transported by vehicles to the quayside. The QCs then remove the containers from the vehicles and load them onto the vessels. The processes for import activities are similar but they are done in the reverse order. For transshipment activities, the processes are a little different. The containers will be stored in the storage yard after they are unloaded from the vessel and will be finally loaded onto other vessels. In this paper, our study is focused on the yard storage allocation problem in a transshipment hub where transshipment of containers is the major activity and the yard activity is intensive. This transshipment port uses prime mover as the main vehicle to transport the containers.

The storage yard plays a pivotal role in transshipment hubs. Most containers unloaded from one vessel are stored in the storage yard and are eventually loaded onto other vessels. Multi-level stacking of containers is a common practice when the volume of transshipment activities is intensive and the land is scarce. This can lead to high concentration of activities within a small area and may likely cause traffic congestion of prime movers. Another result of over stacking is unproductive reshuffles of containers. Traffic congestion and reshuffles can reduce the productivity of resources, which include prime movers, YCs, and QCs. Related works on the yard storage allocation problem is summarized as follows.

The efficiency of stacking depends greatly on the strategies of allocating storage space to arriving containers. Chen (1999) distinguishes several major factors that influence operational efficiency and cause unproductive container movements in terminal operations. Chung et al. (1988) propose the use of buffer space to increase the utilization of the material handling equipment and reduce the total container loading time. Kim and Kim (1999) propose a segregation strategy to allocate storage space for import containers. In Chen et al. (2000) the storage space allocation problem is examined with a time-space network with the objective of allocating containers to storage locations in advance. Taleb-Ibrahimi et al. (1993) describe handling and storage strategies for export containers and quantify their performance according to the amount of space and number of handling moves required. 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 in a bay for a given initial stacking configuration. Kim and Bae (1998) discuss how to reshuffle export containers in container terminals. Kim et al. (2000) propose a methodology to determine the storage location of an arriving export container by considering its weight. Kim and Park (2003) discuss how to allocate storage space for outbound containers that will arrive at a storage yard. Zhang et al. (2003) study the storage space allocation problem in the storage yard of terminals. Chen et al. (1995), Davies and Bischoff (1999), and Scheithauer (1999) study a strategy called consignment. This strategy attempts to store containers with the same destination, contents, and loading time together in some dedicated storage area. Crainic et al. (1993), Cheung and Chen (1998), and Shen and Khoong (1995) look into the yard storage allocation problem for empty containers.

From the literature, it can be seen that various problems associated with yard operations have been addressed. However, these papers do not sufficiently address

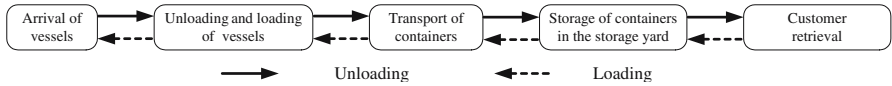


Fig. 2 A flow diagram demonstrating the interaction between container terminal processes

the particular needs of transshipment hubs, but more on the general terminals which emphasize on import and export activities. For transshipment hubs, the loading and unloading activities are both concentrated and need to be considered at the same time. This makes the planning problem much more challenging compared to port planning for general terminals where the loading and unloading activities can be considered independently by having different dedicated storage areas for import and export activities.

If the yard storage allocation problem is not handled properly at a transshipment yard, the problems of reshuffling and traffic congestion might arise. In the port that we study in this paper, the consignment strategy is used. This strategy stores export and transshipment containers with the same departing vessel together at the same designated storage areas. Import containers are not considered in this study as the port operator does not mix the storage area of the import containers with the export and transshipment containers. This strategy helps to reduce the reshuffling level to a negligible level. To handle the traffic congestion, a new workload balancing protocol is proposed and the details will be discussed in [Problem description](#).

The rest of this paper is organized as follows. [Problem description](#) provides the detailed description of the problem. [Model development](#) describes the model development. Numerical experiments and the computational results are presented in [Numerical experiments](#). Two heuristic algorithms are proposed and implemented in [Heuristic algorithms](#). [Conclusions and future research](#) give conclusions and some future research topics.

2 Problem description

The port that we are studying handles a high volume of transshipment containers. One of its competitive edges is that the port is able to turn around the vessels within a short time. At the planning stage, this translates to the requirement for the port operator to allocate the incoming transshipment and export containers to the storage yard such that the traffic congestion can be kept at a minimal level.

To manage the yard allocation process more efficiently, the port operator organizes the storage yard into several blocks as shown in [Fig. 3](#). The depth of each block is six containers and the length of the block is 40 containers. Every block is further divided into five subblocks, where the length of each sub-block is eight containers. The stacking height is five containers (which we call tier). The basic unit for the yard storage allocation process is at the subblock level, i.e., for the consignment strategy, we will assign the containers that are going to the same departing vessel to the same subblocks. There is an assigned lane for the movement of containers by prime movers (the “truck path”) and a separate “passing” lane strictly to allow trucks to pass each other when required. The passing lane is narrow, and it is shared by yard cranes.

Traffic congestion may happen when too much workload needs to be handled within a small area at the same time. For example, if there are a lot of container

movements in subblocks 7 and 12 (see Fig. 3), then there will be many prime movers waiting or moving nearby. This will cause traffic congestion. Similarly, if the workload in subblocks 6 and 7 is high, then the prime movers waiting at subblock 7 may block other prime movers from going to subblock 6 as they share the same path.

To ensure smooth flow of traffic, the port operator has imposed several restrictions during the planning stage. Among them are:

1. When a subblock is in the loading process, its neighboring subblocks should not have any loading or unloading activities.
2. There should not be two or more neighboring subblocks which are having high unloading activities.

To incorporate these restrictions into the planning model, we introduce a “high workload” rule/protocol and a vicinity matrix.

The protocol of high-low workload is to ensure that at any given time, many yard cranes will be highly utilized as the jobs are concentrated as they do not need to move around frequently to other subblocks to perform jobs. The ranges of high workload and low workload do not overlap. For example, the range of high workload is set between 50 and 100 containers per shift, while the low workload is set between 0 and 20 containers per shift by the port operator.

To capture the possible traffic congestion between subblocks, we use a vicinity matrix to represent the neighborhood structure between different sub-blocks. A sub-block is a neighbor if it is adjacent. Adjacency of subblocks inherently implies that trucks must use the same path. For example, subblock 7 is said to be a neighbor of subblocks 6 and 12. Subblock 7 is not a neighbor of subblock 2 even though they

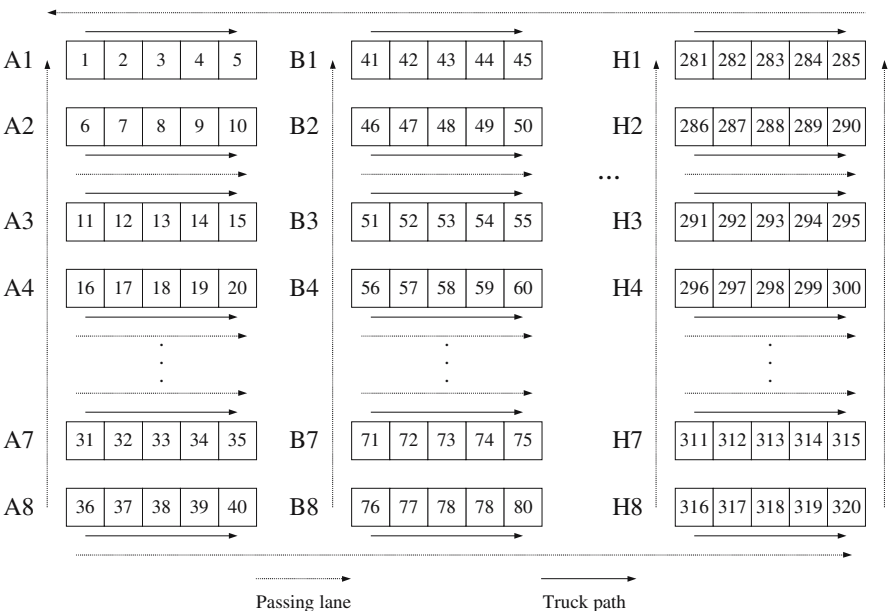


Fig. 3 A typical yard configuration

3.1 Modeling assumptions

The following assumptions are made when developing the storage allocation model.

1. The time span of the model is 7 days with three planning periods in each day. Each planning period corresponds to an 8-h working shift.
2. We assume that the amount of containers arriving in every shift is given and will repeat weekly (this implies the planning period can be wrapped around). The actual number can vary, but for planning purposes, it is reasonable to assume that it is deterministic and an input to the model.
3. At any given time, if the subblock is during the loading activity, then, a dedicated yard crane will be assigned to that subblock as loading activities have higher priority.
4. Two different types of containers are handled in a container terminal. They are 20 ft containers and 40 ft containers. As one subblock consists of a few lanes, it is possible to have a mixture of different types of containers in one subblock. For simplicity, we assume they can be stored in the same subblocks. However, the model can easily be modified to handle the dedicated subblock for different types of containers.
5. All containers that arrive in a given shift will be stored in a subblock until they are loaded onto the departing vessel. The loading activities at any subblock have to be completed within two shifts because this is a requirement set by the port operator to reflect its current service level.
6. A yard crane assigned to a particular block should work until the end of the shift.

3.2 Notations

The model parameters are as follows:

I	The number of sub-blocks under consideration
J	The number of vessels under consideration in the planning horizon
K	The number of blocks under consideration
T	The number of shifts under consideration
N_i	The set of subblocks that are neighbors of subblock i , $1 \leq i \leq I$
V_j	The set of sub-blocks that are reserved for vessel j , $1 \leq j \leq J$
B_k	The set of subblocks that belong to block k , $1 \leq k \leq K$
WX_{jt}	The number of 20-ft containers of the departing vessel, j , which arrive in shift t . It is given and treated as input of the model, $1 \leq j \leq J$, $1 \leq t \leq T$
WY_{jt}	The number of 40-ft containers of the departing vessel, j , which arrive in shift t . It is given and treated as input of the model, $1 \leq j \leq J$, $1 \leq t \leq T$
CS	The capacity of each subblock in terms of TEUs, which is 240 (5 tiers \times 8 lanes \times 8 slots) in this model
CC	The capacity of each yard crane in terms of container moves per shift, which is 100 in this model
C_k	The maximum number of yard cranes allowed to reside in block k at any one time, $1 \leq k \leq K$

NL _{kt}	The number of subblocks in loading process in block k in shift t , $1 \leq k \leq K$, $1 \leq t \leq T$
HL	The lowest value that a high workload can take
HU	The highest value that a high workload can take
LL	The lowest value that a low workload can take
LU	The highest value that a low workload can take

Subscript, i , is for subblock, j , for vessel, k , for block, t , for shift
 The decision variables are as follows:

x_{it}	The number of 20-ft containers that are allocated to subblock, i , for unloading in shift t , $1 \leq i \leq I$, $1 \leq t \leq T$
y_{it}	The number of 40-ft containers that are allocated to subblock, i , for unloading in shift t , $1 \leq i \leq I$, $1 \leq t \leq T$
h_{it}	$=1$ means that the total workload ($x_{it}+y_{it}$) that are allocated to subblock, i , for unloading in shift, t , is high, that is $H_l \leq x_{it}+y_{it} \leq H_u$, $1 \leq i \leq I$, $1 \leq t \leq T$ $=0$ means that the total workload ($x_{it}+y_{it}$) that are allocated to subblock, i , for unloading in shift t is low, that is $L_l \leq x_{it}+y_{it} \leq L_u$, $1 \leq i \leq I$, $1 \leq t \leq T$
d_{kt}	The number of yard cranes allocated to block, k , for unloading in shift t , $1 \leq k \leq K$, $1 \leq t \leq T$

3.3 Model formulation

Managers in container terminals always attempt to reduce costs by efficiently utilizing resources, including berths, storage yards, quay cranes, yard cranes, prime movers, and human resources. In the proposed model, the total number of crane shifts required to handle all the workload should be minimized. It is not to determine the necessary number of yard cranes that should be available in the storage yard for the terminals. However, the operating cost should be reduced by putting less yard cranes in use. Each crane shift corresponds to one active yard crane in one shift. The model is formulated as follows.

$$(SAP) \text{Min } w = \sum_{k=1}^K \sum_{t=1}^T d_{kt} \quad (1)$$

Subject to:

$$\sum_{i \in V_j} x_{it} = WX_{jt} \text{ For all } 1 \leq j \leq J, 1 \leq t \leq T \quad (2)$$

$$\sum_{i \in V_j} y_{it} = WY_{jt} \text{ For all } 1 \leq j \leq J, 1 \leq t \leq T \quad (3)$$

$$\sum_{t=1}^T (x_{it} + 2y_{it}) \leq CS \text{ For all } 1 \leq i \leq I \quad (4)$$

$$\sum_{t=1}^T (x_{it} + y_{it}) \leq 2CC \text{ For all } 1 \leq i \leq I \quad (5)$$

$$\sum_{i \in B_k} (x_{it} + y_{it}) \leq d_{kt} CC \text{ For all } 1 \leq k \leq K, 1 \leq t \leq T \quad (6)$$

$$HL + (LL - HL)(1 - h_{it}) \leq x_{it} + y_{it} \leq LU + (HU - LU)h_{it} \text{ For all } 1 \leq i \leq I, 1 \leq t \leq T \quad (7)$$

$$x_{i't} = 0, y_{i't} = 0 \text{ For all } i' \in N_i, t \in L_i, 1 \leq i \leq I \quad (8)$$

$$\sum_{i' \in N_i \text{ or } i'=i} h_{i't} \leq 1 \text{ For all } 1 \leq i \leq I, 1 \leq t \leq T \quad (9)$$

$$d_{kt} + NL_{kt} \leq C_k \text{ For all } 1 \leq k \leq K, 1 \leq t \leq T \quad (10)$$

$$x_{it} \geq 0, y_{it} \geq 0 \text{ For all } 1 \leq i \leq I, 1 \leq t \leq T \quad (11)$$

$$h_{it} \in \{0, 1\} \text{ For all } 1 \leq i \leq I, 1 \leq t \leq T \quad (12)$$

$$d_{kt} \text{ Integer For all } 1 \leq i \leq I, 1 \leq k \leq K, 1 \leq t \leq T \quad (13)$$

Constraints 2 and 3 ensure that all the workload unloaded from each vessel in each shift will be allocated to corresponding storage locations. Constraint 4 ensures the capacity restriction of subblocks. Constraint 5 ensures that the containers in each subblock should be loaded to the vessels in a certain time span (two shifts in the proposed model). Constraint 6 ensures that the yard cranes for unloading in each block can handle all the unloading workload in each shift.

A subblock with high unloading workload can share a yard crane with neighbors with low unloading workload. Also, several subblocks with low unloading workload can share a yard crane. To make full use of yard cranes, the workload allocated to each subblock in each shift should be either high or low. In this model, constraint 7 is used to ensure this restriction. Constraint 8 ensures that all the neighbors of a loading subblock cannot accept any containers arriving in that shift. Constraint 9 ensures that high unloading workload cannot be allocated to two subblocks that are neighbors to each other in the same 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. One yard crane is required for each loading subblock, and hence, the number of loading subblocks is exactly equal to the number of yard cranes assigned

to that block for loading. Constraint 10 ensures this restriction. Constraints 11, 12 and 13 are nonnegative and integer restrictions.

The model is not easy to solve because the MIP structure is poor and there are too many integer and binary variables. In the next section, the problem will be solved with CPLEX 8.1, a commercial software package.

4 Numerical experiments

In this section, the proposed storage allocation model (SAP) is first tested using two sets of input data for the simplified small-scale problem. Then it is used to solve a large-scale problem close to a moderate terminal.

4.1 Small-scale problem experiment

4.1.1 Input data

For the small-scale problem, there are eight blocks arrayed in four rows and two columns in the storage yard which is shown in Fig. 4. A vicinity matrix can be determined easily from the yard configuration.

It is assumed that there is exactly one vessel being loaded in each shift. The lowest and highest value that a high workload can take are 100 and 200, respectively, and those for a low workload are 0 and 100, respectively. The capacity of 1-yd crane is 200 container moves per shift. The maximum number of yard cranes that can reside in each block is two at any one time. Only five shifts are considered in the model. The parameters are drawn from the real practice with some minor changes to make the model feasible.

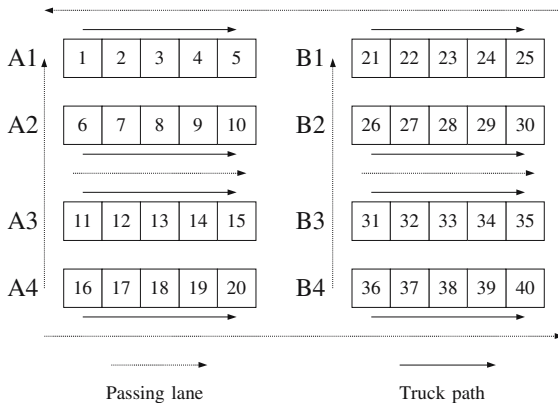


Fig. 4 Yard configuration for the small-scale problem

4.1.2 Implementation

The proposed model is implemented in C++ and run on a Pentium IV computer (CPU: 2.4 GHz, Memory: 512 M). The mixed-integer programming model is solved using CPLEX 8.1 with concert technology, a commercial software package with C++ optimization modeling library and interface.

The computational results are summarized in Tables 2 and 3. Two sets of input data are used to conduct the small-scale problem experiments. The utilization is defined as the ratio of the total storage space occupied by all the unloading containers to the total storage space in the storage yard.

As shown in Tables 2 and 3, for relatively low utilization scenarios (with utilization less than 0.45), the optimal solution can be obtained easily. For moderate utilization scenarios (with utilization between 0.45 and 0.60), it will take a longer time to solve as there are too many choices to allocate containers to their destination storage locations. In addition, too many choices can result in a very large branch and bound tree, which may cause the computer to run out of memory (for example, case 1 with utilization of 0.45). For high utilization scenarios (with utilization greater than 0.90 for case 1 and 0.70 for case 2), the problem is prone to be infeasible as the capacity constraints cannot be satisfied. The different results indicate that the proposed model is data dependent.

Table 2 Results of SAP for small-scale problem (case 1)

Utilization	Computation time (s)	Solution status	Objective Value (# of YCs)
0.05	1	Optimal	10
0.10	1	Optimal	10
0.15	1	Optimal	10
0.20	1	Optimal	10
0.25	1	Optimal	10
0.30	274	Optimal	15
0.35	340	Optimal	15
0.40	1	Optimal	15
0.45	34,196.03	Out of memory	17.56 (LB=20, gap=12.18%)
0.50	3,833	Optimal	20
0.55	110	Optimal	21
0.60	13,304	Optimal	25
0.65	114	Optimal	25
0.70	6	Optimal	25
0.75	11,602	Optimal	30
0.80	6	Optimal	30
0.85	328	Optimal	35
0.90	16	Optimal	35
0.95	1	Infeasible	–
1.00	1	Infeasible	–

Table 3 Results of SAP for small-scale problem (case 2)

Utilization	Computation time (s)	Solution status	Objective value (# of YCs)
0.05	1	Optimal	10
0.10	1	Optimal	10
0.15	1	Optimal	10
0.20	1	Optimal	11
0.25	1	Optimal	12
0.30	1	Optimal	13
0.35	2	Optimal	15
0.40	91	Optimal	17
0.45	1	Optimal	18
0.50	936	Optimal	21
0.55	643	Optimal	23
0.60	106	Optimal	23
0.65	36	Optimal	27
0.70	26	Optimal	28
0.75	1	Infeasible	–
0.80	1	Infeasible	–
0.85	1	Infeasible	–
0.90	1	Infeasible	–
0.95	1	Infeasible	–
1.00	1	Infeasible	–

4.2 Large-scale problem experiment

4.2.1 Input data

For the large-scale problem, there are 64 blocks arrayed in eight rows and eight columns. The layout of the yard configuration of the experiment problem is shown in Fig. 3. The time horizon is 7 days with three shifts in each day. The scale of the problem (around five million TEUs) is comparable to a section of the terminal that we are investigating.

The lowest and highest value that a high workload can take are 50 and 100, respectively, and those for low workload are 0 and 20, respectively. The capacity of 1-yd crane is 100 container moves per shift. The maximum number of yard cranes that can reside in each block is two at any one time. The parameters are fairly reflective on any terminal with high traffic intensity.

4.2.2 Implementation

The large-scale problem is also implemented in C++ and run on the same computer as that for the small-scale problems. The computational results are summarized in Table 4.

As shown in Table 4, the large-scale problem cannot be solved to optimality in a reasonable time and always terminates as a result of insufficient memory. The results just before running out of memory are presented. In addition, for scenarios

with utilization between 0.3 and 0.35, it turns out to be infeasible. This is due to constraints 7, which restricts the workload allocated to each subblock to be either high or low. For some input data, it is impossible to satisfy these constraints.

For such a large-scale problem, the proposed MIP model (in total there are 7,392 integer variables and 24,370 constraints) is too complex to solve to optimality in a reasonable time. Therefore, heuristic algorithms should be developed to find a good enough solution to meet the requirement of port operators. To evaluate the performances of the heuristics, it is necessary to find a lower bound.

4.2.3 Finding a lower bound

One possible way to find a lower bound of this model is to solve each shift independently. Under this assumption, constraints 4 and 5 can be removed from the formulation. SPP is used to name the model to find a lower bound that is mentioned above. It is shown as follows.

$$(SPP) \text{ Min } w = \sum_{k=1}^K d_{kt} \tag{14}$$

Table 4 Results of model SAP for the large-scale problem

Utilization	Computation time (s)	Solution status	Result details
0.05	30,346.80	Out of memory	104, gap=13.69% (lb=89.758)
0.10	24,640.89	Out of memory	149, gap=36.07% (lb=95.257)
0.15	26,680.84	Out of memory	188, gap=36.34% (lb=119.68)
0.20	25,721.92	Out of memory	217, gap=36.09% (lb=138.68)
0.25	64,013.97	Out of memory	221, gap=33.27% (lb=147.47)
0.30	1	Infeasible	–
0.30<U<0.35	1	Infeasible	–
0.35	1	Infeasible	–
0.40	11,060.56	Out of memory	285, gap=20.18% (lb=227.50)
0.45	11,037.89	Out of memory	315, gap=18.86% (lb=255.58)
0.50	13,642.97	Out of memory	352, gap=18.88% (lb=285.53)
0.55	16,114.41	Out of memory	364, gap=14.00% (lb=313.04)
0.60	25,609.03	Out of memory	396, gap=13.57% (lb=342.27)
0.65	23,034.78	Out of memory	415, gap=10.80% (lb=370.17)
0.70	26,500.50	Out of memory	453, gap=11.96% (lb=398.84)
0.75	58,407.64	Out of memory	486, gap=11.71% (lb=429.09)
0.80	220,047.20	Out of memory	524, gap=12.69% (lb=457.53)
0.85	196,890.58	Out of memory	557, gap=12.79% (lb=485.74)
0.90	554,958.83	Out of memory	605, gap=14.93% (lb=514.69)
0.95	684,439.48	Out of memory	623, gap=12.82% (lb=543.12)

Subject to:

$$\sum_{i \in V_j} x_{it} = WX_{jt} \text{ For all } 1 \leq j \leq J \quad (15)$$

$$\sum_{i \in V_j} y_{it} = WY_{jt} \text{ For all } 1 \leq j \leq J \quad (16)$$

$$\sum_{i \in B_k} (x_{it} + y_{it}) \leq d_{kt} CC \text{ For all } 1 \leq k \leq K \quad (17)$$

$$H_l + (L_l - H_l)(1 - h_{it}) \leq x_{it} + y_{it} \leq L_u + (H_u - L_u)h_{it} \text{ For all } 1 \leq i \leq I \quad (18)$$

$$x_{i't} = 0, y_{i't} = 0 \text{ For all } i' \in N_i, t \in L_i, 1 \leq i \leq I \quad (19)$$

$$\sum_{i' \in N_i \text{ or } i'=i} h_{i't} \leq 1 \text{ For all } 1 \leq i \leq I \quad (20)$$

$$d_{kt} + L_{kt} \leq C_k \text{ For all } 1 \leq k \leq K \quad (21)$$

$$x_{it} \geq 0, y_{it} \geq 0 \text{ For all } 1 \leq i \leq I \quad (22)$$

$$h_{it} \in \{0, 1\} \text{ For all } 1 \leq i \leq I \quad (23)$$

$$d_{kt} \text{ Integer For all } 1 \leq k \leq K \quad (24)$$

The objective is to minimize the total number of yard cranes used in the current shift t . All constraints ensure the same restrictions as the original model, SAP, within one shift.

The SPP model is also implemented in C++ and run on the same computer as that for the original model, SAP. For comparison the same input data as those for both the small-scale problem and the large-scale problems are used. The computational results are presented in Tables 5, 6 and 7.

As shown in Table 5, model SPP for the first case small-scale problem gives good lower bounds for most scenarios. Most of them have the same objective value as the optimal solution. As model SPP for each shift can be solved efficiently, the lower bound for the original model SAP can be obtained within a short time. As shown in Table 6, for the second case small-scale problem, the lower bound for every scenario is exactly the same as the objective value of the optimal solution.

For the large-scale SAP model, the optimal solution cannot be obtained. Instead, only the best lower bound before running out of memory can be obtained. As shown in Table 7, the lower bound obtained from model SPP is always better than the best lower bound obtained from the model SAP before running out of memory.

5 Heuristic algorithms

The model is intractable when the size of the problem becomes large. In this section, two heuristic algorithms that may find a feasible solution close to the optimal solution in a reasonable time are proposed.

5.1 The sequential method

It can be seen from the procedure that a lower bound is found wherein the single shift model can be solved effectively. Inspired by this finding, the proposed model can be solved one shift at a time. This is called the sequential method (SQM). The difference between SPP and SQM is that in the sequential method, the linking constraints of Eqs. 4 and 5 will be considered so as to ensure that the solution is feasible. Specifically, a sequence of shifts should be picked first, and based on this sequence, the model is solved shift by shift. Note that after solving the model for each shift, the remaining capacity of each subblock should be updated. Therefore, constraints 4 and 5 can be modified as follows

$$x_{it} + 2y_{it} \leq CS - \sum_{\tau \in \Gamma_t} (x_{i,\tau} + 2y_{i,\tau}) \text{ For all } 1 \leq i \leq I \tag{25}$$

$$x_{it} + y_{it} \leq CC - \sum_{\tau \in \Gamma_t} (x_{i,\tau} + y_{i,\tau}) \text{ For all } 1 \leq i \leq I \tag{26}$$

where Γ_t is the set which consists of all those shifts that come before the current shift, t , in a given sequence. Constraints 25 and 26 ensure the capacity restriction of each subblock in terms of storage space and yard cranes, respectively. Hence, the SQM is the same as the procedure of SPP except that the two additional constraints, Eqs. 25 and 26 are added into SPP. The sequential method may be effective as it has much less decision variables and constraints.

It is also noted that the sequence of shifts chosen need not be in chronological order as the demand repeats weekly and the only linking constraints are the space

Table 5 Comparison of results of SAP and SPP for small-scale problem (case 1)

Utilization	Results of SAP	Lower bounds from SPP	
		Computation time (s)	Objective value
0.05	10	1	10
0.10	10	1	10
0.15	10	1	10
0.20	10	1	10
0.25	10	1	10
0.30	15	1	15
0.35	15	1	15
0.40	15	1	15
0.45	17.56 (lb=20, gap=12.18%)	1	20
0.50	20	1	20
0.55	21	1	20
0.60	25	1	25
0.65	25	1	25
0.70	25	1	25
0.75	30	1	30
0.80	30	1	30
0.85	35	1	35
0.90	35	1	35

Table 6 Comparison of results of SAP and SPP for small-scale problem (case 2)

Utilization	Results of SAP	Lower bounds from SPP	
		Computation time (s)	Objective value
0.05	10	1	10
0.10	10	1	10
0.15	10	1	10
0.20	11	1	11
0.25	12	1	12
0.30	13	1	13
0.35	15	1	15
0.40	17	1	17
0.45	18	1	18
0.50	21	1	21
0.55	23	1	23
0.60	23	1	23
0.65	26	1	26
0.70	28	1	28

restriction and crane capacity restriction. It does not matter which shift we choose first, as it will not violate any of these constraints. On the other hand, different solutions are obtained from different shift orderings which implies that the sequence is important. However as there are $T!$ different sequences that can be

Table 7 Comparison of results of SAP and SPP for large-scale problem

Utilization	Lower bounds from SAP before MR		Lower bounds from SPP	
	Lower bound	Computation time (s)	Lower bound	Computation time (s)
0.05	89.758	30,346.80	102	21
0.10	95.257	24,640.89	138	744
0.15	119.68	26,680.84	174	5,217
0.20	138.68	25,721.92	196	8,706
0.25	147.47	64,013.97	196	19,603
0.30	Not found	2	Not found	6,229
0.35	Not found	2	Not found	1,297
0.40	227.50	11,060.56	239	337
0.45	255.58	11,037.89	266	297
0.50	285.53	13,642.97	296	523
0.55	313.04	16,114.41	323	134
0.60	342.27	25,609.03	353	56
0.65	370.17	23,034.78	381	32
0.70	398.84	26,500.50	409	27
0.75	429.90	58,407.64	438	152
0.80	457.53	220,047.20	468	66
0.85	485.74	196,890.58	495	27
0.90	514.69	554,958.83	524	53
0.95	543.12	684,439.48	556	25

used, and it is time consuming to enumerate all of them, we propose to consider a subset of sequences which is as follows: select any shift in the planning horizon as the start shift of the sequence. Then starting from this shift, the sequence is formed by including the shifts according to the order of time. When the end of the planning horizon is reached, the sequence is wrapped around the beginning of the planning horizon, and it is continued until it reaches the shift immediately before the start shift.

5.2 The column generation method

Another heuristic algorithm is the column generation method (CGM). To be consistent, the same notations are used as those defined in the original model, SAP. In addition, some new notations are defined to represent the column coefficients and new decision variables.

In the column generation model, a column represents a storage allocation in a particular shift. The column coefficients represent the workload assigned to each subblock, the number of yard cranes required and the high-low pattern for that shift. They are listed below.

- x_{itr} The number of 20-ft containers allocated to subblock, i , in shift, t , for column, r , $1 \leq i \leq I$, $1 \leq t \leq T$
- y_{itr} The number of 40-ft containers allocated to subblock, i , in shift, t , for column, r , $1 \leq i \leq I$, $1 \leq t \leq T$
- h_{itr} =1 if the workload allocated ($x_{itr}+y_{itr}$) to subblock, i , in shift, t , for column, r , is high, i.e., $H_l \leq x_{itr}+y_{itr} \leq H_u$, $1 \leq i \leq I$, $1 \leq t \leq T$
 =0 if the workload allocated ($x_{itr}+y_{itr}$) to subblock, i , in shift, t , for column, r , is low, i.e., $L_l \leq x_{itr}+y_{itr} \leq L_u$, $1 \leq i \leq I$, $1 \leq t \leq T$
- d_{ktr} The number of yard cranes allocated to block, k , in shift, t , for column, r , $1 \leq k \leq K$, $1 \leq t \leq T$
- n_{tr} The total number of yard cranes required in shift, t , for column, r , $1 \leq t \leq T$
- p_{tr} =1, if column, r , is selected for shift t
 =0, otherwise.

In the column generation model, decision variable, p_{tr} , is used to represent whether column, r , is selected for shift t ; it should be binary in the original MIP master problem, while it should be continuous between 0 and 1 in the relaxed master problem.

In the master problem, the total number of crane shifts required for the whole planning horizon under consideration should be minimized.

$$(CGM) \text{Min } w = \sum_{t=1}^T \sum_r n_{tr} p_{tr} \tag{27}$$

Subject to:

$$\sum_r p_{tr} = 1 \text{ For } 1 \leq t \leq T \tag{28}$$

$$\sum_{t \notin L_i} \sum_r (x_{itr} + 2y_{itr})p_{tr} \leq CS \text{ For } 1 \leq i \leq I \tag{29}$$

$$\sum_{t \notin L_i} \sum_r (x_{itr} + y_{itr})p_{tr} \leq 2 \text{ CC For } 1 \leq i \leq I \tag{30}$$

$$p_{tr} \in \{0, 1\} \text{ For } 1 \leq t \leq T, \forall r \tag{31}$$

Constraint 28 ensures that one and only one column can be selected in each feasible solution for each shift. Constraint 29 and 30 ensure the capacity restriction of subblocks in terms of storage space and yard cranes. Constraint 31 is integer restrictions.

To solve the column generation model, the master problem should be feasible, and all the variables should be continuous, which are necessary to obtain the dual prices information. Therefore, in the relaxed master problem, p_{tr} is a continuous variable assuming a value between 0 and 1. A feasible solution to the original problem is added as the initial columns. This can ensure the feasibility of the relaxed master problem.

After obtaining the dual prices from the relaxed master problem, they can be used to price out the new columns. A new column with the most negative objective function coefficient in the master problem can be found from the pricing problem for each shift. After adding the new columns, the master problem can be solved again to obtain the updated dual prices.

The pricing problem is defined as follows:

$$\text{Min } z = n_{tr} - \pi_t - \sum_{i=1}^I (x_{itr} + 2y_{itr})\sigma_i - \sum_{i=1}^I (x_{itr} + y_{itr})\delta_i \tag{32}$$

where π_t , σ_i , and δ_i are dual prices for constraints 28, 29 and 30, respectively. The objective function of the pricing problem is to find the most negative objective function coefficient in the master problem.

Subject to:

$$\sum_{i \in V_j} x_{itr} = WX_{jt} \text{ For } 1 \leq j \leq J \tag{33}$$

$$\sum_{i \in V_j} y_{itr} = WY_{jt} \text{ For } 1 \leq j \leq J \tag{34}$$

$$\sum_{i \in B_k} (x_{itr} + y_{itr}) \leq d_{kt} \text{ CC For } 1 \leq k \leq K \tag{35}$$

$$H_l + (L_l - H_l)(1 - h_{itr}) \leq x_{itr} + y_{itr} \leq L_u + (H_u - L_u)h_{itr} \text{ For } 1 \leq i \leq I \quad (36)$$

$$x_{i'tr} = 0, y_{i'tr} = 0 \text{ For } 1 \leq i \leq I, i' \in N_i, t \in L_i \quad (37)$$

$$\sum_{i' \in N_i \text{ or } i'=i} h_{i'tr} \leq 1 \text{ For } 1 \leq i \leq I \quad (38)$$

$$d_{ktr} + L_{kt} \leq C_k \text{ For } 1 \leq k \leq K \quad (39)$$

$$\sum_{k=1}^K d_{ktr} = n_{tr} \quad (40)$$

$$x_{itr} \geq 0, y_{itr} \geq 0 \text{ For } 1 \leq i \leq I, 1 \leq t \leq T, \forall r \quad (41)$$

$$h_{itr} = 0 \text{ or } 1 \text{ For } 1 \leq i \leq I, 1 \leq t \leq T, \forall r \quad (42)$$

$$n_{tr} \geq 0 \text{ Integer For } 1 \leq t \leq T, \forall r \quad (43)$$

All constraints ensure the same restrictions as the original model, SAP, within one shift. Constraints 4 and 5 are removed from the formulation as only one shift is considered at one time.

Table 8 Results of heuristic algorithms for small-scale problem (case 1)

Utilization	Results of SAP	Results of SQM	Results of CGM
0.05	10	10	10
0.10	10	10	10
0.15	10	10	10
0.20	10	10	10
0.25	10	10	10
0.30	15	15	15
0.35	15	15	15
0.40	15	15	15
0.45	17.56 (gap=12.18%)	20	20
0.50	20	20	20
0.55	21	21	21
0.60	25	25	25
0.65	25	26	26
0.70	25	Not Found	38
0.75	30	Not Found	40
0.80	30	Not Found	40
0.85	35	Not Found	40
0.90	35	Not Found	40
>=0.95	Infeasible	Infeasible	Infeasible

To accelerate the column generation procedure, the results obtained from model SPP and SQM are treated as alternative columns. In conjunction with the initial feasible solution, the column generation method may improve the quality of the results obtained from the sequential method.

5.3 Implementation

The sequential method and the column generation method are both implemented in C++ and run on the same computer as that for the original model, SAP. They use the same input data as those for the original model, SAP. The computational results are presented in Tables 8, 9 and 10.

Tables 8 and 9 present the results of the sequential method and the column generation method for small-scale problems. For relatively low utilization scenarios, the obtained feasible solutions are quite close to the optimal solutions. For some scenarios, the objective value of the feasible solution obtained from SQM is exactly the same as that of the optimal solution. For relatively high utilization scenarios, the sequential method may not find any feasible solution. It is because the sequential method is a greedy algorithm and cannot ensure feasibility.

For relatively low utilization scenarios, the column generation method can yield near-optimal or optimal solutions as the results of model SQM are considered as alternative columns. For relatively high utilization scenarios, there may be a big gap between the solutions obtained and the lower bounds.

Table 10 compares the results of the sequential method and the column generation method for large-scale problem. For most relatively low and moderate utilization scenarios (with utilization up to 0.80 except between 0.3 and 0.35), the sequential method can give near-optimal or exact the optimal solution. This is also true for the column generation method that uses the results of the sequential method as alternative columns. However, for very high utilization scenarios (with

Table 9 Results of heuristic algorithms for small-scale problem (case 2)

Utilization	Results of SAP	Results of SQM	Results of CGM
0.05	10	10	10
0.10	10	10	10
0.15	10	10	10
0.20	11	11	11
0.25	12	12	12
0.30	13	13	13
0.35	15	15	15
0.40	17	17	17
0.45	18	18	18
0.50	21	21	21
0.55	23	23	23
0.60	23	24	24
0.65	26	Not Found	40
0.70	28	Not Found	40
≥ 0.75	Infeasible	Infeasible	Infeasible

Table 10 Results of heuristic algorithms for large-scale problem

Utilization	Lower bounds from SPP	Results of SQM	Results of CGM
0.05	102	102	102
0.10	138	138	138
0.15	174	174	174
0.20	196	196	196
0.25	196	197	197
0.30	Infeasible	Infeasible	Infeasible
0.35	Infeasible	Infeasible	Infeasible
0.40	239	239	239
0.45	266	266	266
0.50	296	296	296
0.55	323	323	323
0.60	353	353	353
0.65	381	381	381
0.70	409	409	409
0.75	438	438	438
0.80	468	472	468
0.85	495	Not Found	2,048
0.90	524	Not Found	2,048
0.95	556	Not Found	2,048
1.00	584	Not Found	2,048

utilization greater than 0.8), the sequential method does not work well. Instead, the column generation method can give a feasible solution.

In the experiments conducted, the column generation method gives a better solution than that of the sequential method only for those problems wherein the sequential method can get a feasible answer. This means that the column generation model actually doesn't improve the solution quality effectively. Even though the relaxed master problem can be solved to optimality easily, the columns generated cannot improve the quality of solution to the MIP version master problem. For each single shift, there are too many optimal solutions as there are too many possible allocations with the same number of cranes. Once the master problem finds some linear combination of the columns that can satisfy the linking constraints, the column generation procedure will stop right away. Consequently, the solutions for different shifts will not be explored enough. We can use some meta-heuristics to generate more columns for a future research topic.

6 Conclusions and future research

In this paper, we study an actual problem faced by a leading transshipment port operator. Currently, the port operator uses consignment strategy and attempt to assign containers to different subblocks with the objective of preventing traffic congestion. However, they do not have any formal planning tool and its decisions are based on intuition and past experiences. Hence, we develop a tool that is able to provide a holistic and systematic way to address the problem which takes into

consideration the port operator's actual requirements. The model is based on a MIP formulation and is particularly useful for transshipment hubs where transshipment of containers is the major activity, and the yard activity is intensive. Although the model under some scenarios cannot be solved to optimality by the commercial software package, we have developed two heuristics based on the MIP model to provide good results. While the heuristic algorithms cannot guarantee an optimal solution, we have developed a bound which is useful in quantifying the quality of these solutions. So far this is the first paper to address the yard allocation problem with consignment strategy and vicinity matrix for a transshipment port.

In the current model, we assume that the yard template is given, i.e., the set of subblocks which is assigned to a certain departing vessel is known. Given this information, we determine where we should allocate the containers in each shift. A future research direction is to look at how to design the yard template and integrate this to the current yard storage allocation model. Another possible future research topic is to identify good ways to solve the MIP model efficiently, which include the integration of meta-heuristics with the MIP model, a different decomposition method which can exploit the structure of the MIP model.

References

- Chen CS, Lee SM, Shen QS (1995) An analytical model for the container loading problem. *Eur J Oper Res* 80(1):68–76
- Chen CY, Chao SL, Hsieh TW (2000) A time-space network model for the space resource allocation problem in container marine transportation. Paper presented at the 17th international symposium on mathematical programming, 2000, Atlanta, USA
- Chen T (1999) Yard operations in the container terminal—a study in the “unproductive moves”. *Marit Policy Manage* 26(1):27–38
- Cheung RK, Chen CY (1998) A two-stage stochastic network model and solution methods for the dynamic empty container allocation problem. *Transp Sci* 32(2):142–162
- Chung YG, Randhawa SU, McDowell ED (1988) A simulation analysis for a transtainer-based container handling facility. *Comput Ind Eng* 14(2):113–115
- Crainic TG, Gendreau M, Dejax P (1993) Dynamic and stochastic models for the allocation of empty containers. *Oper Res* 41:102–126
- Davies AP, Bischoff EE (1999) Weight distribution considerations in container loading. *Eur J Oper Res* 114: 509–527
- Kim KH (1997) Evaluation of the number of reshuffles in storage yards. *Comput Ind Eng* 32(4):701–711
- Kim KH, Bae JW (1998) Re-marshaling export containers in port container terminals. *Comput Ind Eng* 35:655–658
- Kim KH, Kim HB (1999) Segregating space allocation models for container inventories in port container terminals. *Int J Prod Econ* 59:415–423
- Kim KH, Park KT (2003) A note on a dynamic space-allocation method for outbound containers. *Eur J Oper Res* 148(1):92–101
- Kim KH, Park YM, Ryu KR (2000) Deriving decision rules to locate export containers in storage yards. *Eur J Oper Res* 124:89–101
- Ryan NK (1998) The future of maritime facility designs and operations. In: *Simulation Conference 1998*, pp 1223–1227

-
- Scheithauer G (1999) LP-based bounds for the container and multi-container loading problem. *Int Trans Oper Res* 6(2):199–213
- Shen WS, Khoong CM (1995) A DSS for empty container distribution planning. *Decis Support Syst* 15:75–82
- Taleb-Ibrahimi M, De Castilho B, Daganzo CF (1993) Storage space vs handling work in container terminals. *Transp Res B* 27:13–32
- Vis IFA, De Koster R (2003) Transshipment of containers at a container terminal: an overview. *Eur J Oper Res* 147(1):1–16
- Yun WY, Choi YS (1999) A simulation model for container-terminal operation analysis using an object-oriented approach. *Int J Prod Econ* 59(1–3):221–230
- Zhang C, Liu J, Wan Y-w, Murty KG, Linn RJ (2003) Storage space allocation in container terminals. *Transp Res B* 37:883–903