Transportation Research Part E 48 (2012) 516–528

Contents lists available at SciVerse ScienceDirect

Transportation Research Part E

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

Terminal and yard allocation problem for a container transshipment hub with multiple terminals

Der-Horng Lee *, Jian Gang Jin, Jiang Hang Chen

Department of Civil and Environmental Engineering, National University of Singapore, Singapore

ARTICLE INFO

Article history:
Received 14 February 2011
Received in revised form 16 June 2011
Accepted 31 July 2011

Keywords:
Terminal allocation
Yard allocation
Container ports
Transshipment

ABSTRACT

This paper presents an integer programming model for the terminal and yard allocation problem in a large container transshipment hub with multiple terminals. The model integrates two decisions: terminal allocation for vessels and yard allocation for transshipment container movements within a terminal as well as between terminals. The objective function aims to minimize the total inter-terminal and intra-terminal handling costs generated by transshipment flows. To solve the problem, we develop a 2-level heuristic algorithm to obtain high quality solutions in an efficient way. Computational experiments show the effectiveness of the proposed approach.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

In container transshipment hubs, the management of transshipment flows is an important issue to which port operators pay close attention. Transshipment containers are temporarily stored in storage yards after being discharged from inbound vessels, and wait to be loaded onto outbound vessels in the near future. This transshipment movement generates container flows between quay side and yard side. As transshipment containers do not need to move out of the terminal gates, the related operations concentrate on storage yards and along the quay. Consequently, management for transshipment flows, including berth allocation, yard allocation and so on, is required to achieve a high productivity.

The Port of Singapore is one of the world's busiest transshipment hubs and handles one-fifth of the world's total transshipment throughput. Along with the increase of containerized maritime shipping, the Port of Singapore has set up five terminals phase by phase and another one is under construction in order to meet the increasing demand. It is often the case that a large transshipment hub consists of several terminals which are close to each other. Fig. 1 presents such a multi-terminal transshipment system with three terminals in Singapore.

For such a multi-terminal system where many handling resources are involved, operations are complex and there are some unique issues calling for attention which are different from traditional ones in the management of a single terminal. One problem comes from inter-terminal traffic and it is what the port operators concern most. This is because inter-terminal traffic contributes to the whole operational cost to a large extent. In the case that two related vessels berth at two different terminals, for example, T1 and T3 in Fig. 1, there exists an inter-terminal container movement operation which needs a lot of resources including yard cranes and yard trucks. When inter-terminal traffic volume becomes high, not only does cost increase, traffic congestion may also occur. Take Fig. 1 as an example, there is only one traffic corridor indicated by the dotted arrows between T1 and T3 and high traffic could lead to high costs and traffic congestion. Fortunately, inter-terminal traffic

* Corresponding author. Address: Block E1A #07-16, 1 Engineering Drive 2, National University of Singapore, Singapore 117576, Singapore. Tel.: +65 6516 2131; fax: +65 6779 1635.

E-mail address: dhl@nus.edu.sg (D.-H. Lee).

1366-5545/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.tre.2011.09.004

Fig. 1. A multi-terminal system in Singapore.

could be reduced by assigning related vessels to the same terminal as long as enough berth capacity is available. Hence, terminal allocation for such a multi-terminal system deciding the visiting terminal for each vessel should be carefully planned in order to reduce inter-terminal traffic.

Another issue is to allocate yard storage space and to manage container transshipment flows within yards through their duration-of-stay. It is referred to as yard allocation problem in this paper. Storage areas need to be allocated before containers are discharged from inbound vessels. Before loading operation, containers should be moved to a yard which is close to the berth position of the corresponding outbound vessel in order to speed up loading. Hence, a reallocation is needed to move containers between the two allocated yards, especially when the first assigned yard is far from the berthing position of the outbound vessel. Such container flows between quay side and yard side as well as between yards result in yard crane operation cost and yard truck transportation cost. In a transshipment hub where storage areas are scarce, the management of container flows plays an important role in reducing the operational costs. Yard allocation studied in this paper concerns not only the assignment of storage resource for incoming containers but also the reallocation of yards to manage inter-terminal and intra-terminal container flows at different time periods. A reallocation conducted between yards inside a terminal and between terminals causes intra-terminal cost and inter-terminal cost, respectively. A good yard allocation plan generates low intra-terminal as well as inter-terminal costs.

As the above two problems, terminal allocation and yard allocation, could affect the operational costs significantly in a transshipment hub, we develop an integrated model for the terminal and yard allocation problem at a tactical level trying to minimize the handling cost of the transshipment flows. Our motivation in addressing this terminal and yard allocation problem from a tactical viewpoint is to help port operators improve the management of such a multi-terminal system and achieve competitive operational costs.

2. Literature review

In open literature, plenty of research has studied berth allocation problem (BAP) and yard allocation problem (YAP). For BAP, the basic task is to assign berth resource to incoming vessels at certain time with specific objectives. BAP can be categorized into two types: discrete BAP and continuous BAP in terms of the management of berth resource. Imai et al. (2001) address the problem of dynamic berth allocation where berth resource is discretized. The objective of the problem is to minimize the sum of waiting and handling times for every ship. In Guan and Cheung (2004), the continuous BAP is studied with the objective of minimizing total weighted turnaround time. In the discrete case, a berth could only accommodate one vessel at a time and vessel size is not considered. However, vessels can berth at any position along the quay in the continuous case and vessel size is considered. A lot of other works extend their study and we refer readers to Bierwirth and Meisel (2010) for more information about BAP. Traditional BAP considers the situation at the operational level where the planning horizon is short and the exact calling schedule of incoming vessels is known to the port. For long term berth allocation, Giallombardo et al. (2010) develop a model which integrates berth allocation and quay crane assignment at a tactical level. By assigning berth and quay crane resources, the authors try to maximize the total value of chosen quay crane profiles (i.e. no. of quay cranes per working shift) and at the same time minimize the housekeeping costs generated by transshipment flows between vessels. Hendriks et al. (2011) study a multi-terminal container port and address the problem of spreading a set of cyclically calling vessel lines over different terminals and allocating a berthing and departure time to each vessel. The objective is to reduce the amount of inter-terminal container movement and to balance the quay crane workload over the terminals and over time. Our research resembles that of Hendriks et al. (2011) as we both consider the terminal allocation for a multi-terminal container port instead of assigning the exact berth locations within a container terminal. However, they include the consideration of quay crane workload while we consider the storage yard allocation for transshipment flows. For transshipment terminals with limited storage yards, yard allocation should be planned very carefully since the management of transshipment flows inside yards within a terminal and between terminals determines the operational costs to a large extent.

Storage yard allocation problem deals with determining the storage position in the yards and the amount of storage space to allocate for incoming containers. In Kim and Kim (2002), two cost models are presented to decide optimal amount of storage space and optimal number of transfer cranes for handling import containers under different circumstances. Kim and Park (2003) develop a mixed integer linear programming model for pre-allocating storage space for arriving outbound containers in order to utilize space efficiently and to achieve maximum efficiency of loading operation. Zhang et al. (2003) study the storage space allocation problem with a hierarchical approach in a container terminal where import, export and transshipment containers are mixed in storage blocks. The problem is decomposed into two levels and each level is formulated as a mathematical optimization model. The first level is to balance the workload among all blocks in order to reduce berthing time. The second level is to minimize the total distance between storage blocks and vessel berthing locations by allocating containers to the storage space determined in the first level. The above literature either focuses on import and export containers or does not differentiate the types of containers. However, the different characteristics of transshipment flow make the above methods inapplicable for transshipment hubs. To the best of our knowledge, the literature about transshipment-related problems is very scarce. Moccia and Astorino (2007) present a problem called *Group Allocation Problem* considering the transshipment flow in the yards through the duration-of-stay period. A mathematical model is formulated with the objective of minimizing all the handling costs generated by discharging, loading and reallocation of container groups. However, their work applies to the single terminal operation (only intra-terminal transshipment flow cost is considered) and assumes that the berth allocation plan is given.

In this paper, on one hand we extend the study by Moccia and Astorino (2007) to a multi-terminal circumstance. Our aim is to manage the transshipment container flow and to reduce inter-terminal and intra-terminal transportation costs. On the other hand, as terminal allocation largely affects the inter-terminal traffic, we also include the decision of terminal allocation. Hence, we study the terminal and yard allocation problem in a transshipment hub with multiple terminals from a tactical point of view. Compared with existing literature, the advantages of our study are:

- We study the container transshipment flow management problem in a multi-terminal transhipment hub to include the consideration of the inter-terminal container movement rather than only focus on the optimization of a single terminal.
- An integrated terminal allocation and yard allocation model is presented for a multi-terminal transshipment system so as to achieve a more effective management of container flows through a port.

The following paper is organized as follows: problem description and mathematical model is presented in Section 3. Section 4 provides the heuristic approach, followed by numerical experiments in Section 5. At last, Section 6 draws the conclusion.

3. Mathematical model

3.1. Problem description

This paper is to study the tactical terminal and yard allocation problem (TYAP) for a container transshipment hub which consists of several terminals located close to each other. The terminal allocation problem is to allocate the visiting terminal for each calling vessel satisfying the calling schedules requested by shipping liners. The objective of the problem is to minimize the total inter-terminal transportation cost which is incurred by inappropriate terminal allocation. More specifically, if two vessels berth at different terminals and there is a group of containers exchanged between them, the containers should be moved between the two berthing terminals. This reallocation of storage yard results in inter-terminal transportation cost. Another problem is related to yard management which is to allocate and reallocate storage yards to containers. It is to decide in detail when and where to conduct inter-terminal and intra-terminal container reallocations. Fig. 2 shows three cases of transshipment flows in a transshipment hub with three terminals. Case 1 is a transshipment flow without any reallocation as both the inbound and outbound vessels berth at Terminal 1. However, the transshipment flow in Case 3 requires an inter-terminal movement as the two connecting vessels are serviced at different terminals. The transshipment flow in Case 2 has an intra-terminal reallocation which is conducted to move the containers to a yard closer to the quayside for the sake of fast loading operation.

With a discrete planning horizon, we aim to, on one hand, assign a terminal to each vessel, on the other hand to determine the storage allocation plan for transshipment containers with the objective of minimizing the inter-terminal and intra-terminal transportation costs. Containers exchanged between two vessels are treated as a group. Hence, a group is a set of containers sharing the same inbound and outbound vessels as well as the schedule. In this paper, we focus on container groups rather than individual containers. Before presenting the mathematical model, some assumptions are made as follows:

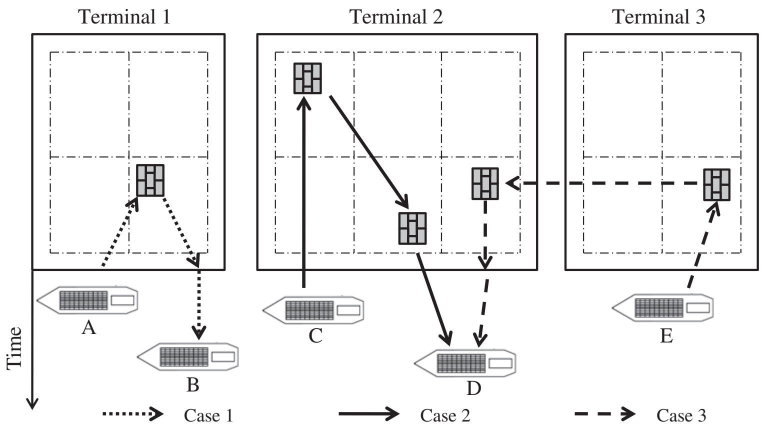

Fig. 2. Three cases of transshipment flows in a transshipment hub with three terminals.

(1) The calling schedule of all vessels is known to the terminal operator.
(2) The exchanging container volume between vessels is assumed to be known.
(3) The discharging and loading operation of one container group can be finished within one time period.

The information in Assumptions (1) and (2) could be obtained from shipping liners and past data because the calling schedule is usually regular and the exchanging container volume is stable within a relatively long period. When the data varies, the TYAP should be updated. The assumptions are used to get the data of the container groups, i.e. the arrival and departure times, the volume of groups. Assumption (3) is reasonable because a container vessel usually carries/receives multiple container groups and the service time of one container group is shorter than the turnaround time of the vessel. In case that the service of a vessel takes longer than one period, the arrival/departure time of the corresponding container groups can be assigned uniformly within the whole service time. For example, if a vessel requires two time periods for discharging and loading, we can assign half of the corresponding container groups to arrive/departure at the first time period, and the rest half to the second time period.

3.2. Model formulation

The TYAP can be considered as a network optimization problem with temporal and spatial dimensions. We define a graph $G_k(N,A)$ for container Group k as depicted in Fig. 3. The source node and sink node are labeled S_k and T_k, respectively. Group k arrives in one terminal at time period a_k and leaves from one terminal at b_k. Through its duration-of-stay from a_k to b_k, there are $\overline{m}$ candidate storage yards to be considered for storing the group at each time period. A path from the source node to sink

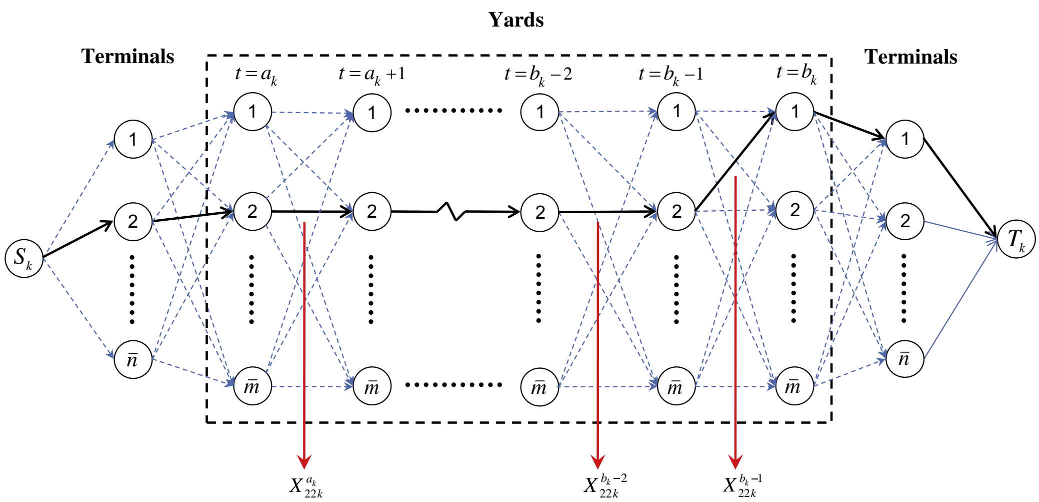

Fig. 3. A graph representation of TYAP for Group k.

node corresponds to a terminal and yard allocation plan for the group. The dotted arrows between terminals and yards are arcs representing movements between quay side and yard side while those inside yards represent reallocation between successive time periods. The arc costs depend on the pair of linked nodes. A path indicated by the solid arrows in Fig. 3 shows a feasible terminal and yard allocation plan for the group. As illustrated, Group k arrives at Terminal 2 and leaves from Terminal 1. Yard 2 is allocated to Group k after unloaded from its inbound vessel. Group k remains in Yard 2 until a reallocation to Yard 1 at time period b_k.

Indices:	
i, j	the index for storage yards and terminals
k	the index for container groups
t	the index for time periods

Parameters:	
M	set of storage yards, $M = \{1, 2, \ldots, \bar{m}\}$
N	set of terminals, $N = \{1, 2, \ldots \bar{n}\}$
V	set of vessels, $V = \{1, 2, \ldots, \bar{v}\}$
T	set of time periods
K	set of container groups
a_k	arrival time of Group k, $a_k \in T$
b_k	departure time of Group k, $b_k \in T$
q_k	storage space requirement of Group k
o_k	inbound vessel of Group k
d_k	outbound vessel of Group k
r_k	the maximum allowed number of reallocations between yards for Group k
Q_i^1	the storage capacity of Yard i, $i \in M$
Q_i^2	the processing capacity of Terminal i in a time period (i.e. the largest amount of containers that can be discharged or loaded at Terminal i), $i \in N$
α_{kl}	set to 1 if Group k and l have the same inbound vessel (i.e. $o_k = o_l$), and 0 otherwise, $k, l \in K$
β_{kl}	set to 1 if Group k and l have the same outbound vessel (i.e. $d_k = d_l$), and 0 otherwise, $k, l \in K$
γ_{kl}	set to 1 if the inbound vessel of Group k and the outbound vessel of Group l are the same (i.e. $o_k = d_l$), and 0 otherwise, $k, l \in K$
δ	the maximum allowed travel cost between quay side to yard side
c_{ij}^1	travel cost between Yard i and Yard j, $i, j \in M$
c_{ij}^2	travel cost between Terminal i and Yard j, $i \in N, j \in M$

The two parameters r_k and δ reflect the managerial practice about container movements. Larger r_k allows more flexibility for port operators to conduct container relocation from one yard to another, but generates more handling cost. δ is introduced to limit the distance between quay side and the container storage location as large distance would slow down the quayside operation. We remark that the cost parameters c_{ij}^1 and c_{ij}^2 actually include two parts: inter-terminal cost and intra-terminal cost. For c_{ij}^1, when Yard i and j are located within the same terminal the cost is the intra-terminal handling cost, while Yard i and j belong to different terminals the cost reflects the inter-terminal handling cost. It is similar for c_{ij}^2. Note that cost parameters can be estimated based on the geographical locations and port operators' experiences.

Decision variables:	
X_{ijk}^t	set to 1 if Group k is located at Yard i at time period t and located at Yard j at time period $t+1$, and 0 otherwise, $i, j \in M$, $k \in K$, $t \in T$
U_{ijk}^t	set to 1 if Group k uses arc $i \to j$ at time period t upon arrival, and 0 otherwise, $i \in N$, $j \in M$, $k \in K$, $t \in T$
V_{ijk}^t	set to 1 if Group k uses arc $i \to j$ at time period t upon departure, and 0 otherwise, $i \in M$, $j \in N$, $k \in K$, $t \in T$
W_{ikt}^1	set to 1 if Group k is located at Yard i at time period t, and 0 otherwise, $i \in M$, $k \in K$, $t \in T$
W_{ikt}^2	set to 1 if Group k is processed (i.e. loaded or discharged) at Terminal i at time period t, and 0 otherwise, $i \in N$, $k \in K$, $t \in T$
Z_{ik}^1	set to 1 if Group k uses Terminal i upon arrival, and 0 otherwise, $i \in N$, $k \in K$
Z_{ik}^2	set to 1 if Group k uses Terminal i upon departure, and 0 otherwise, $i \in N$, $k \in K$

Objective function:

$$\min\left\{\sum_{k\in K}\sum_{i\in M}\sum_{j\in M}\sum_{t\in T}c^1_{ij}q_kX^t_{ijk}+\sum_{k\in K}\sum_{i\in N}\sum_{j\in M}\sum_{t\in T}c^2_{ij}q_k\left(U^t_{ijk}+V^t_{jik}\right)\right\} \tag{1}$$

The objective function consists of two parts as indicated in (1). The first part reflects the inter-terminal and intra-terminal handling costs resulted from reallocation through the duration-of-stay in storage yards. The other part represents the transportation cost between quay side and yard side during discharging and loading operations.

Constraints:

$$\sum_{i\in N}Z^1_{ik}=1 \quad \forall k\in K \tag{2}$$

$$\sum_{i\in N}Z^2_{ik}=1 \quad \forall k\in K \tag{3}$$

$$\sum_{i\in M}X^t_{ijk}=\sum_{i\in M}X^{t+1}_{jik} \quad \forall k\in K, j\in M, \quad a_k\leqslant t\leqslant b_k-2 \tag{4}$$

$$Z^1_{ik}=\sum_{j\in M}U^t_{ijk} \quad \forall i\in N, k\in K, t=a_k \tag{5}$$

$$Z^2_{ik}=\sum_{j\in M}V^t_{jik} \quad \forall i\in N, k\in K, t=b_k \tag{6}$$

$$\sum_{i\in N}U^t_{ijk}=\sum_{i\in M}X^t_{jik} \quad \forall j\in M, k\in K, t=a_k \tag{7}$$

$$\sum_{j\in N}V^t_{ijk}=\sum_{j\in M}X^{t-1}_{jik} \quad \forall i\in M, k\in K, t=b_k \tag{8}$$

$$\sum_{i\in N}\sum_{j\in M}c^2_{ij}U^t_{ijk}\leqslant\delta \quad \forall k\in K, t=a_k \tag{9}$$

$$\sum_{i\in M}\sum_{j\in N}c^2_{ji}V^t_{ijk}\leqslant\delta \quad \forall k\in K, t=b_k \tag{10}$$

$$W^1_{ikt}=\sum_{j\in M}X^t_{ijk} \quad \forall i\in M, k\in K, a_k\leqslant t\leqslant b_k-1 \tag{11}$$

$$W^1_{ikt}=\sum_{j\in N}V^t_{ijk} \quad \forall i\in M, k\in K, t=b_k \tag{12}$$

$$W^2_{ikt}=\sum_{j\in M}(U^t_{ijk}+V^t_{jik}) \quad \forall i\in N, k\in K, t\in T \tag{13}$$

$$\sum_{k\in K}q_kW^1_{ikt}\leqslant Q^1_i \quad \forall i\in M, t\in T \tag{14}$$

$$\sum_{k\in K}q_kW^2_{ikt}\leqslant Q^2_i \quad \forall i\in N, t\in T \tag{15}$$

$$\sum_{t\in T}\sum_{i\in M}\sum_{j\in M|j\neq i}X^t_{ijk}\leqslant r_k \quad \forall k\in K \tag{16}$$

$$\alpha_{kl}(Z^1_{ik}-Z^1_{il})=0 \quad \forall i\in N, k\in K, l\in K, k\neq l \tag{17}$$

$$\beta_{kl}(Z^2_{ik}-Z^2_{il})=0 \quad \forall i\in N, k\in K, l\in K, k\neq l \tag{18}$$

$$\gamma_{kl}(Z^1_{ik}-Z^2_{il})=0 \quad \forall i\in N, k\in K, l\in K, k\neq l \tag{19}$$

$$X^t_{ijk}, U^t_{ijk}, V^t_{ijk}, Z^1_{ik}, Z^2_{ik}, W^1_{ikt}, W^2_{ikt}\in\{0,1\} \tag{20}$$

Constraints (2)–(8) are the flow conservation constraints. Constraints (2) show the outflow requirement at source node while Constraints (3) ensure the inflow at sink node. By the two constraints, each group has an inbound terminal and an outbound terminal and this indirectly assigns visiting terminals for calling vessels. Flow conservation inside yards is ensured by Constraints (4). The relationship between decision variable **Z** and **U**,**V** is indicated by Constraints (5) and (6) which link terminal allocation and loading, discharging operation decisions. Similarly, Constraints (7) and (8) deal with the relationship between decision variable **U**, **V** and **X**. Constraints (9) and (10) specify the travel cost requirement between quay side and yard side in order to ensure a fast loading and discharging operation. Constraints (11)–(13) define the decision variable **W** which represents container locations through duration-of-stay. At any time period, the total storage space requirement of all the groups in the same yard should not exceed the yard storage capacity, as ensured by Constraints (14). Similarly, terminal capacity is guaranteed by Constraints (15) as the total amount of loading and discharging containers should be within the processing capacity of the terminal. Constraints (16) guarantee the number of reallocations through duration-of-stay inside yards of each group respects the maximum allowed reallocation times. Terminal allocation constraints for groups of the same inbound/outbound vessels are indicated by (17)–(19) since such groups can only serviced at the same terminal. Finally, Constraints (20) specify the domain for decision variables.

4. Heuristic approach

Consider the problem with a given terminal allocation plan, the TYAP is reduced to the *Group Allocation Problem* (Moccia and Astorino, 2007) which is proved to be NP-hard by the authors. Hence, the TYAP is also NP-hard and generally it is difficult and not efficient to solve the problem by a commercial solver especially for large scale problems. Hence, we propose a 2-level heuristic approach to find good solutions within a short computational time. The framework of the heuristic is introduced in Section 4.1 and the details are presented in the Sections 4.2 and 4.3.

4.1. Framework of the heuristic

In this heuristic, the terminal and yard allocation problem is solved hierarchically as illustrated by the heuristic flowchart in Fig. 4. The heuristic consists of two levels. Level 1 is designed to obtain a good terminal allocation plan for vessels. In this level, we solve the linear programming relaxation of the remaining yard allocation problem in order to evaluate the fitness of terminal allocation plans efficiently. Neighborhood search technique is employed to find better solutions in the searching process. At the end of Level 1, a good terminal allocation plan is obtained and passed onto Level 2. In Level 2, the terminal allocation plan is treated as input information and yard allocation for container groups is determined in detail. Such a subproblem in Level 2 is actually equivalent to *Group Allocation Problem* (Moccia and Astorino, 2007). We develop a tabu search based heuristic method to find good yard allocation plans and obtain the total inter-terminal and intra-terminal handling costs.

4.2. Level 1

In Level 1, we focus on the terminal allocation problem and try to obtain a good terminal allocation plan. Firstly, an initial solution for terminal allocation problem is randomly generated. Then, neighborhood search is conducted in the neighborhood of the current solution. In each searching loop, NS_1 neighborhood solutions are generated. With a given terminal allocation plan, the related decision variables can be easily derived, i.e. Z_{ik}^1, Z_{ik}^2, W_{ikt}^2. To evaluate the fitness of the neighborhood

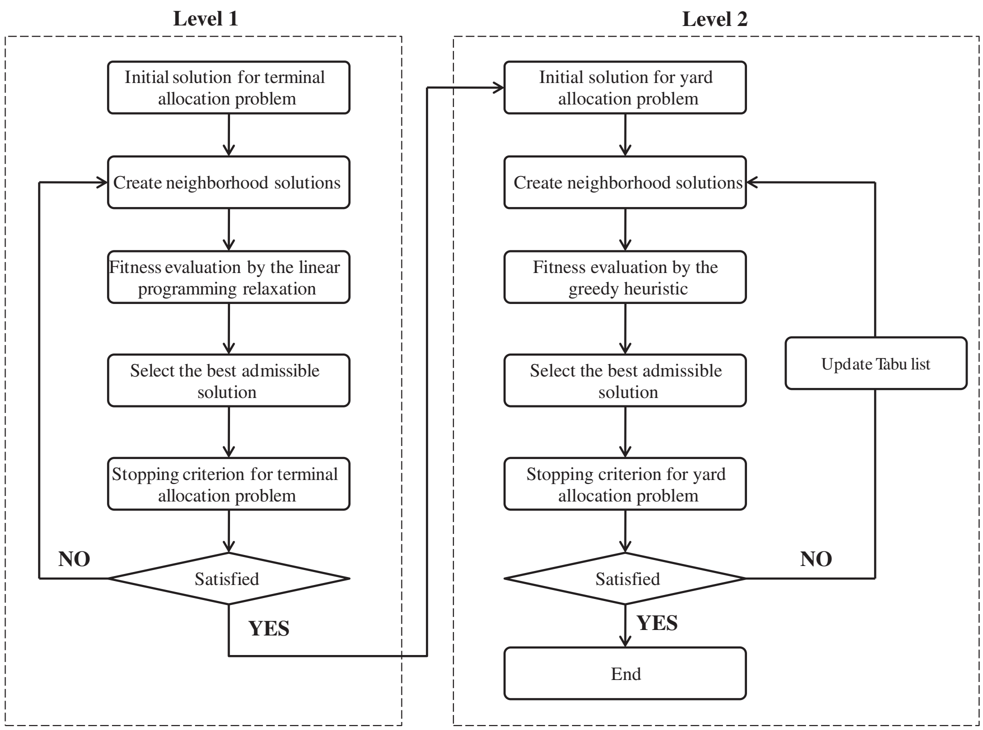

Fig. 4. The flowchart of the 2-level heuristic framework.

solutions, the remaining yard allocation problem is solved approximately by relaxing the integer constraints of the yard allocation decision variables. At the end of each searching loop, the best admissible solution is selected to update the current solution. The stopping criterion is also checked to decide whether to continue the neighborhood search process or to end the first level. When the stopping criterion is met, the best solution in the whole searching process is selected as the terminal allocation plan and passed onto Level 2 as input information. The details are illustrated as follows:

4.2.1. Encoding representation

We apply a straightforward encoding representation for the terminal allocation problem. Let $S = (s_1, s_2, \cdots, s_i, \cdots, s_{\bar{v}})$ represents the terminal allocation decisions for $\bar{v}$ vessels where $s_i \in N$ indicates the calling terminal of Vessel *i*. For example, $S = (1,1,3,2,3)$ is a terminal allocation solution for five vessels. As indicated by the solution, Vessel 1 calls at Terminal 1, Vessel 4 is served at Terminal 2 and so on for the other three vessels.

4.2.2. Neighborhood structure

For the neighborhood search phase, two patterns of neighborhood structure as shown in Fig. 5 are employed: pair-wise interchange and flipping patterns. In Pattern 1, two components of the solution are randomly selected and interchanged with each other. Pattern 2 only conducts operation on one component. The position of the component is randomly generated and the component s_i randomly flips to $s_j \in N \backslash \{s_i\}$. As illustrated by Fig. 5, the terminal positions of Vessel 2 and 4 are interchanged with each other in Pattern 1. The terminal position of Vessel 3 is switched from 3 to 2 in Pattern 2. The two patterns of neighborhood search are conducted with an equal probability.

4.2.3. Fitness evaluation

With a candidate terminal allocation solution, the decision variables Z^1_{ik}, Z^2_{ik} and W^2_{ikt} can be easily derived. However, it is difficult to solve the remaining problem and obtain the exact total inter-terminal and intra-terminal costs since the yard allocation problem is still an NP-hard integer programming problem. In order to evaluate terminal allocation solutions, the linear programming relaxation technique is applied by replacing the integer constraint that the decision variables must be 0 or 1 by a weaker constraint that they belong to the interval [0,1]. The relaxed linear program of the remaining yard allocation problem is as follows:

$$\begin{aligned} &\min && (1) \\ &\text{s.t.} && (4) - (16) \\ & && X^t_{ijk}, U^t_{ijk}, V^t_{ijk}, W^1_{ikt} \in [0,\, 1] \end{aligned}$$

It is worth noting that the linear programming relaxation has a physical meaning: container groups can be further divided into smaller sub-groups. Thus, a container group can be stored separately in different yards. Since the storage capacity of a yard or a terminal is relatively larger than the amount of storage space needed by a container group, the solution gap between the integer program and its relaxation is small. Besides, from the viewpoint of computational complexity it is much easier to solve a linear program than an integer program. That is why we apply linear programming relaxation to evaluate the fitness of terminal allocation solutions.

4.2.4. Stopping criterion

The neighborhood search for the terminal allocation problem in Level 1 terminates when the following condition is met: best solution does not change for SC_1 consecutive iterations.

4.3. Level 2

With a given terminal allocation plan for vessels, the arrival and departure positions of groups are known and the remaining problem is to determine the container flows in storage yards within their duration-of-stay. A tabu search based greedy heuristic is developed for Level 2.

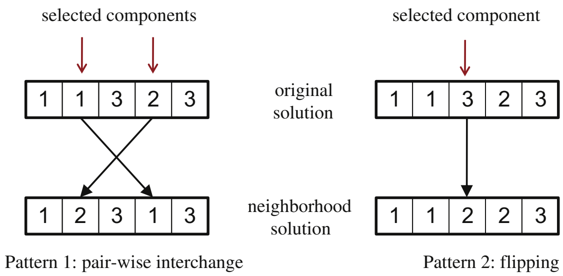

Fig. 5. Two patterns of neighborhood structure.

The main idea of the heuristic is as follow: the storage area is considered as a two dimensional network with limited capacity, i.e. spatial and temporal dimensions. The groups are loaded onto the network sequentially and each group chooses its shortest path (i.e. storage plan with a least cost) in the network. The loading of each group corresponds to a loading stage. At each stage, the capacity of the network is updated after the loading of the corresponding group. With the updated network, the next group finds the shortest path available in the network. When all the groups are loaded onto the network, the complete yard allocation plan is determined and the objective function value can be easily obtained by adding the path costs of all the groups together.

Fig. 6 presents an illustrative example with two yards, three time periods and two groups. For notational convenience, the storage capacity of all the yards is assumed to be the same denoted by Q. Group 1 is firstly loaded onto the network with full capacity and the path with least cost is $S_1 \rightarrow (2,1) \rightarrow (2,2) \rightarrow (1,3) \rightarrow T_1$ which is indicated by solid arrow in Fig. 6a. Then the capacity of the network can be updated and Group 2 is loaded onto the new network as depicted in Fig. 6b. Due to the capacity constraint, there is only one feasible path for Group 2 which is $S_2 \rightarrow (1,2) \rightarrow (2,3) \rightarrow T_2$. Finally, the residual network is shown in Fig. 6c and the objective function value can be easily obtained by summing up the costs of the two paths. It should be noted that different loading sequences for groups result in different solutions and objective values.

The challenge is to determine a good sequence for loading groups onto the network by which near optimal solutions can be obtained. The tabu search technique is employed in this level to find a good loading sequence. A loading sequence determines the order of groups to be loaded onto the network. The groups loaded earlier have a network with larger capacity than latter ones. This implies earlier groups have a higher priority than the latter ones. In each neighborhood search loop, NS_2 neighborhood solutions are generated and evaluated. The details of the heuristic are introduced as follows:

4.3.1. Encoding representation

Let $P = (p_1, p_2, \cdots, p_{|K|})$ be the loading sequence for all the groups where p_i represents the group that are loaded onto the space–time network at the i^{th} order. For example, $P = (2,3,1)$ is an encoding representation for a case with three groups. Group 2 is firstly loaded onto the network with full capacity followed by Group 3. Group 1 is the last one to load onto the network.

4.3.2. Neighborhood structure

To generate neighborhood solutions, only the pair-wise interchange pattern in Fig. 5 is applied.

4.3.3. Fitness evaluation

Given the sequence of groups for loading onto the network, the fitness of the solution could be evaluated by the algorithm in Table 1.

In the fitness evaluation algorithm, a very important step is to find the path with least handling cost for the current group given a network with limited capacity. To find such a path is similar to the classical shortest path problem. However, the unique features of this step are: the number of nodes between source node and sink node is fixed which is determined by the arrival and departure time periods; nodes have limited capacity and the route choice is restricted by the maximum

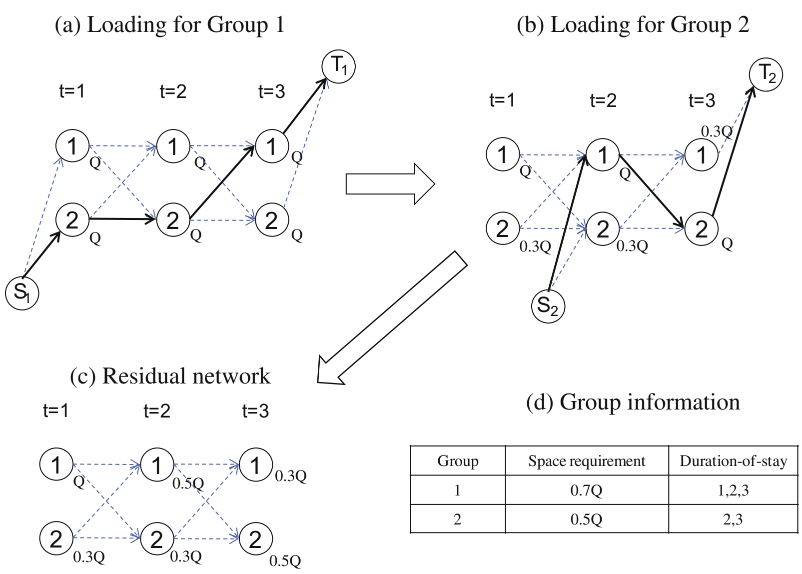

Fig. 6. An illustrative example of the proposed heuristic for the problem in Level 2.

Table 1
The pseudo code for the fitness evaluation.

1:	$P = (p_1, p_2, \ldots, p_{\lvert K \rvert}) \leftarrow$ some candidate solution
2:	Objective_value $\leftarrow$ 0 ►to record the handling cost
3:	Network $\leftarrow Q$ ►to record the network capacity
4:	**for** $k = 1$ to $\lvert K \rvert$ **do**
5:	$PT \leftarrow$ Path (p_k) ► find the path with least cost for Group p_k
6:	Objective_value $\leftarrow$ Objective_value + Cost (PT)
7:	Network $\leftarrow$ Network − Space (PT) ►update network capacity
8:	**end**
9:	**return** Objective_value

number of reallocations allowed. Hence, in order to solve the specific step, we employ the dynamic programming algorithm presented as follows:

- Stage variable: time period $t \in T$
- State variable: node (i,t)
- Optimal value function: $L(n,i,t)$ represents the minimum cost from source node S_k to the current node (i,t) with n times of reallocation prior to the node
- Recursive function:

$$L(n,i,t+1) = \min \begin{Bmatrix} L(n,i,t) \\ L(n-1,j,t) + Cost(j,i) \quad \forall j \in N \cup M, j \neq i \end{Bmatrix} \quad \forall i \in N \cup M, t \in T, a_k \leqslant t < b_k, 0 \leqslant n \leqslant r_k \tag{21}$$

- Boundary condition: $L(n,o_k,a_k) = 0 \ \forall n = 0$ to r_k
- Optimal solution: $\min\{L(n,d_k,b_k) \ \forall n = 0$ to $r_k\}$

The recursive function shows that: if no reallocation is conducted, the cost of node (i,t) is the same as that of node $(i,t-1)$ since no handling operation is conducted. However, when a reallocation is needed, the cost of the latter node is the summation of the cost of the previous node and the arc cost linking the two nodes. We remark that before calculating the cost for the current node the capacity constraint should be checked. Only if the capacity requirement holds, the cost can be calculated by the recursive function. Otherwise, the cost of the current node should be infinite since the storage plan is infeasible. It should be noted that we do not provide the details of how to record the least cost path for the sake of brevity. However, it is simple to accomplish the recording by introducing a variable to record the preceding node on the least cost path for each node. With the least cost path information, the network capacity could be easily updated. In case that no feasible path could be found for certain group, such a solution should be discarded.

4.3.4. Tabu list

In the tabu list, the positions of pair-wise interchange are recorded. First-in-first-out rule is applied to update the tabu list which means the oldest information would be removed out of the list to accommodate new one.

4.3.5. Stopping criterion

The heuristic with local search process in Level 2 will be terminated when the following condition is met: the best solution does not change for SC_2 consecutive iterations.

5. Numerical experiment

In this section, we report the design and results of a comprehensive numerical experiment. The 2-level heuristic is coded in C++ and calls a commercial solver CPLEX 12.1 to solve the relaxed linear program in Level 1. The mathematical model presented in Section 3 is also solved by CPLEX 12.1 to obtain optimal results. A comparison between the results of the heuristic and CPLEX is presented. All the numerical tests are conducted on a PC with 3 GHz CPU and 4 GB RAM. Some parameters for the heuristic are selected by trials and listed as follows:

- $NS_1 = 2log(\bar{v})$ iterations

- SC_1: best solution does not change for 10 consecutive iterations
- $NS_2 = 25$ iterations
- SC_2: best solution does not change for 30 consecutive iterations
- Length of tabu list: 60.

5.1. Test instances

We use 8 h as the length of a time period and set the storage capacity of a yard equal to 1 unit and berth capacity of a terminal to 10 unit for one time period. For experimental purpose, the maximum number of reallocations allowed r_k is set to 1 for all the groups as we intend to obtain good storage plans without too many reallocations. The maximum inflows and outflows of each vessel is set to 5. Then, the test instances are generated as follows:

Step 1: Generate the set of V for vessels whose calling schedules are uniformly distributed in the planning horizon. Go to Step 2.
Step 2: Container group set $K = \emptyset$. Go to Step 3.
Step 3: Generate a candidate container group k with three attributes: inbound vessel o_k, outbound vessel d_k and storage space requirement q_k. q_k is uniformly distributed in (0,0.5). Go to Step 4.
Step 4: If k respects the maximum flow number requirement of each vessel, accept the candidate group and set $K \leftarrow K \cup \{k\}$, go to Step 3. Otherwise, discard the candidate group and go to Step 5.
Step 5: If 1×10^4 consecutive candidate groups are discarded, go to Step 6. Otherwise, go to Step 3.
Step 6: End the instance generation process.

Fifteen-test instances are generated and two port layouts are selected: a smaller one with three terminals and 12 yards (each terminal has 4 yards) and a larger one with 4 terminals and 20 yards (each terminal has 5 yards). Table 2 shows the parameters of the 15 test instances. Instances I1–I9 have a shorter planning horizon with 6 or 9 time periods (2 or 3 days) and deal with less container groups. For instances I10–I15, the planning horizon is 21 (7 days) and more container groups are included.

Table 2
Parameters of the test instances.

Instance	No. of vessels	No. of time periods	No. of groups	Port layout
I1	15	6	34	3×12
I2	15	6	32	3×12
I3	15	6	34	3×12
I4	20	6	49	3×12
I5	20	6	53	3×12
I6	20	6	57	3×12
I7	25	9	70	3×12
I8	25	9	58	3×12
I9	25	9	69	3×12
I10	30	21	89	4×20
I11	30	21	90	4×20
I12	30	21	92	4×20
I13	40	21	123	4×20
I14	40	21	127	4×20
I15	40	21	125	4×20

Table 3
Computational results of CPLEX and the 2-level heuristic.

Instance	CPLEX		2-level heuristic			Gap (%) $\frac{(2)-(1)}{(1)} \times 100$
	Optimal (1)	CPU (s)	Mean (2)	Stdev	CPU (s)	
I1	40.01	169	41.31	0.45	16.3	3.25
I2	50.34	209	51.66	0.54	18.1	2.61
I3	55.07	371	57.40	0.51	14.5	4.22
I4	78.61	5882	81.77	0.64	25.9	4.02
I5	85.53	4942	88.68	0.40	29.4	3.69
I6	87.42	1485	90.75	0.83	33.9	3.80
I7	94.43	18,694	99.33	1.29	71.1	5.19
I8	96.07	28,383	100.54	0.81	63.7	4.65
I9	98.37	30,073	103.44	1.79	76.2	5.16
I10	–	–	157.28	2.99	1532	–
I11	–	–	153.30	2.07	1365	–
I12	–	–	155.25	3.31	1409	–
I13	–	–	213.60	1.25	4118	–
I14	–	–	220.25	1.83	3763	–
I15	–	–	234.30	2.75	4994	–

Table 4
Comparison of the optimization model and a simple planning method.

Instance	Heuristic (1)	Without optimization (2)	Ratio (1)/(2)
I1	41.31	52.13	0.79
I2	51.66	64.51	0.80
I3	57.40	71.77	0.80
I4	81.77	97.46	0.84
I5	88.68	105.56	0.84
I6	90.75	107.18	0.85
I7	99.33	119.83	0.83
I8	100.54	121.21	0.83
I9	103.44	129.66	0.80
I10	157.28	200.62	0.78
I11	153.30	200.22	0.77
I12	155.25	201.68	0.77
I13	213.60	269.53	0.79
I14	220.25	284.96	0.77
I15	234.30	299.94	0.78
Average			0.80

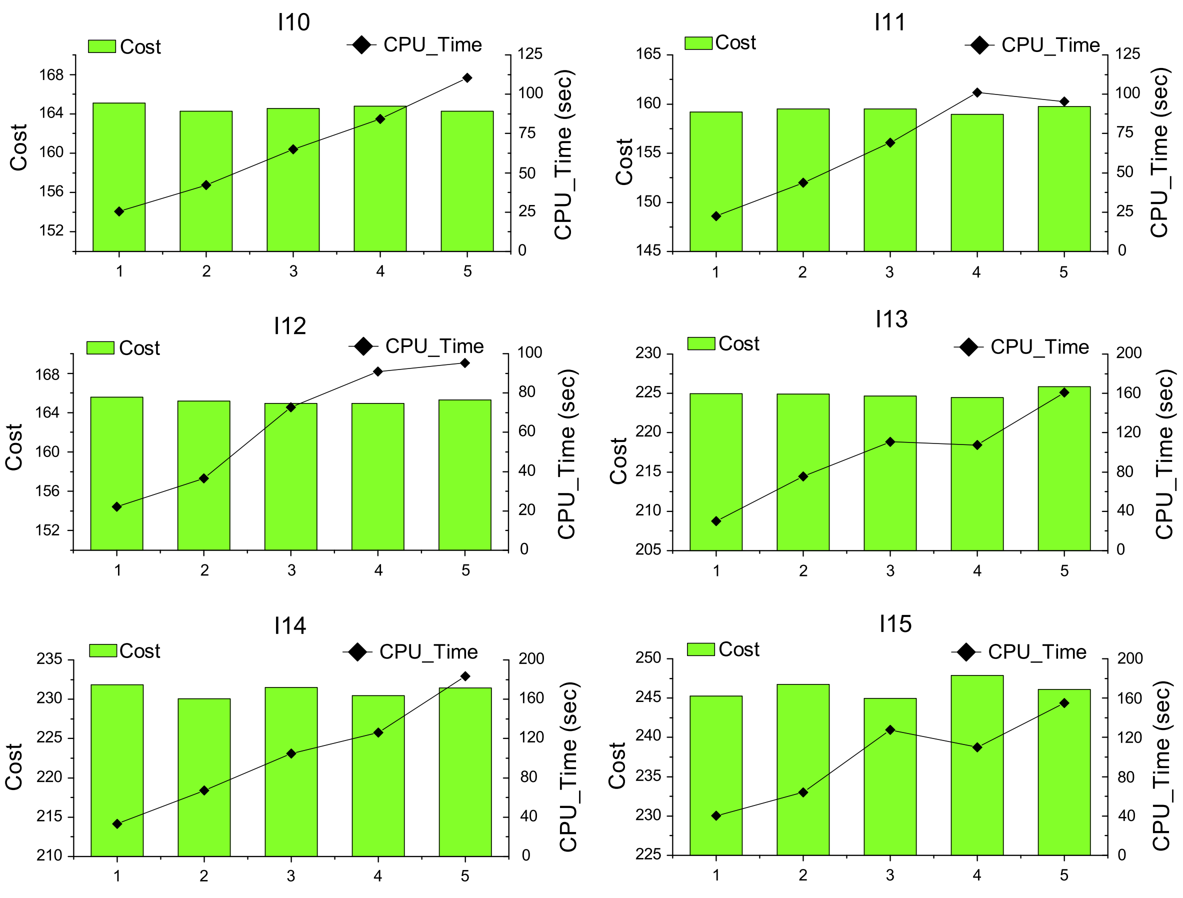

Fig. 7. Sensitivity analysis of the maximum allowed reallocation times r_k.

5.2. Computational results

We compare the optimal results and near-optimal ones obtained from CPLEX and the heuristic, respectively. Due to the randomness in the heuristic searching process, the heuristic may return different results for each run. Hence, we run the heuristic 10 times and report the mean value and standard deviation of the results. For CPLEX, it is terminated when the computational time reaches 12 h or it runs out of memory. Table 3 shows the computational results of CPLEX and the 2-level heuristic for all instances.

As can be seen from Table 3, CPLEX can only handle smaller instances from I1 to I9 and the computational time needed is highly sensitive to the instance scale. However, the 2-level heuristic is able to find high quality solutions within a much shorter computational time. The average gap between the solutions and optimal ones for instance I1–I9 is less than 5% which is acceptable from our point of view. For the larger instances I10–I15, CPLEX cannot even find feasible solutions before it terminates due to out-of-memory while the 2-level heuristic can still handle.

5.3. Optimization improvement

In order to assess the effectiveness of the terminal and yard allocation problem, we compare the result of the 2-level heuristic and that of a simple planning method. This method reflects a simple way to allocate terminals and yards without any optimization consideration: the visiting terminals for vessels are randomly assigned. Upon arrival, each container groups is stored in the yard which is closest to the arrival position of the inbound vessel. Similarly, upon departure each container group is moved to the yard which is closest to the departure position of the outbound vessel. The total inter-terminal and intra-terminal costs of the two planning methods are compared in Table 4.

As can be seen, the TYAP proposed in this paper has a significant improvement (about 20%) over the simple planning method without any optimization consideration. This demonstrates the effectiveness of our optimization model. For the larger instances I10–I15, the improvements are even larger than those of smaller instances which implies the 2-level heuristic also performs well for larger instances.

5.4. Sensitivity analysis

We analyze the performance of the heuristic in Level 2 with different maximum allowed reallocation times r_k as the fitness evaluation step in Level 2 is dependent on r_k. Fig. 7 reports the results and computational times of instances I10–I15 with r_k varying between 1 and 5. For each instance given a certain r_k, firstly ten terminal allocation plans are randomly generated. Then the heuristic in Level 2 is applied to the generated terminal allocation plans. We report the mean value of the objective function values and computational times. The results show a linear relationship between the computational time and the value of r_k. This observation is in line with the computational complexity of the heuristic in Level 2 $O(r_k|M||T||K|)$. Another observation is that increased r_k does not provide significant improvement of the results. Theoretically, the objective value with a larger r_k should be no greater than the one with a smaller r_k. However, the results of the experiment do not conform to this due to the randomness of the heuristic. Overall, $r_k = 1$ is a good compromise between computational effort and solution quality.

6. Conclusion

To conclude, the contributions of the paper to the literature are as follows. First, this paper has extended research on container port operations from a single terminal to a multi-terminal transshipment hub. We have studied from a tactical point of view two practical problems: terminal allocation problem for vessels and yard allocation problem for transshipment container movements. An integer programming model is developed that integrates the two decision problems and aims to minimize total inter-terminal and intra-terminal handling costs generated by transshipment flows. Our study addresses this terminal and yard allocation problem and provides decision support to port operators for the management of transshipment flows in a multi-terminal transshipment hub. Additionally, a 2-level heuristic approach is designed to tackle the integrated problem in an efficient way. Numerical experiments show that the integrated problem can gain significant improvement over a simple management method without any optimization consideration.

References

Bierwirth, C., Meisel, F., 2010. A survey of berth allocation and quay crane scheduling problems in container terminals. European Journal of Operational Research 202 (3), 615–627.
Giallombardo, G., Moccia, L., Salani, M., Vacca, I., 2010. Modeling and solving the tactical berth allocation problem. Transportation Research Part B: Methodological 44 (2), 232–245.
Guan, Y.P., Cheung, R.K., 2004. The berth allocation problem: models and solution methods. OR Spectrum 26 (1), 75–92.
Hendriks, M.P.M., Armbruster, D., Laumanns, M., Lefeber, E., Udding, J.T., 2011. The strategic allocation of cyclically calling vessels for multi-terminal container operators. Flexible Services and Manufacturing Journal, in press.
Imai, A., Nishimura, E., Papadimitriou, S., 2001. The dynamic berth allocation problem for a container port. Transportation Research Part B: Methodological 35 (4), 401–417.
Kim, K.H., Kim, H.B., 2002. The optimal sizing of the storage space and handling facilities for import containers. Transportation Research Part B: Methodological 36 (9), 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 (1), 92–101.
Moccia, L., Astorino, A., 2007. The group allocation problem in a transshipment container terminal. In: Proceedings of World Conference on Transport Research. University of California at Berkeley.
Zhang, C., Liu, J., Wan, Y.-W., Murty, K.G., Linn, R.J., 2003. Storage space allocation in container terminals. Transportation Research Part B: Methodological 37 (10), 883–903.