



## A NOVEL STRATEGY FOR STOWAGE PLANNING OF 40 FEET CONTAINERS IN CONTAINER TERMINALS

Ning Zhao

*Logistics Engineering College, Shanghai Maritime University, 201306, Shanghai, China., ningzhao@shmtu.edu.cn*

Yifan Shen

*Logistics Engineering College, Shanghai Maritime University, 201306, Shanghai, China*

Mengjue Xia

*Logistics Engineering College, Shanghai Maritime University, 201306, Shanghai, China*

Jian Jin

*Logistics Engineering College, Shanghai Maritime University, 201306, Shanghai, China. Taicang Port SP Zhenghe Container Terminals Co. Ltd, Jiangsu Province, China*

Follow this and additional works at: <https://jmstt.ntou.edu.tw/journal>



Part of the [Engineering Commons](#)

### Recommended Citation

Zhao, Ning; Shen, Yifan; Xia, Mengjue; and Jin, Jian (2016) "A NOVEL STRATEGY FOR STOWAGE PLANNING OF 40 FEET CONTAINERS IN CONTAINER TERMINALS," *Journal of Marine Science and Technology*. Vol. 24: Iss. 1, Article 9.

DOI: 10.6119/JMST-016-0125-9

Available at: <https://jmstt.ntou.edu.tw/journal/vol24/iss1/9>

This Research Article is brought to you for free and open access by Journal of Marine Science and Technology. It has been accepted for inclusion in Journal of Marine Science and Technology by an authorized editor of Journal of Marine Science and Technology.

---

# A NOVEL STRATEGY FOR STOWAGE PLANNING OF 40 FEET CONTAINERS IN CONTAINER TERMINALS

## Acknowledgements

This research was supported by “Scientific Research Innovation Project” of Shanghai Municipal Education Commission (No. 14ZZ140), the “Ph. D. Innovation Program” of Shanghai Maritime University (No. 2014ycx040), and the “Yang Fan Plan for Shanghai Youth Science and Technology Talent” of the Science and Technology Commission of Shanghai Municipality (No. 15YF1404900).

# A NOVEL STRATEGY FOR STOWAGE PLANNING OF 40 FEET CONTAINERS IN CONTAINER TERMINALS

Ning Zhao<sup>1</sup>, Yifan Shen<sup>1</sup>, Mengjue Xia<sup>1</sup>, and Jian Jin<sup>1,2</sup>

Key words: container terminal, stowage planning, integer programming, loading efficiency, stowage quality.

## ABSTRACT

Owing that the container terminals increasingly secure a crucial position in today's container transportation, the stowage planning, which is one of the important process during container-loading operations, gradually attracts the attention of terminal operators. In this paper, we discuss the vessel stowage planning problem for 40 feet outbound containers, in which a strategy named 'ROIR' is analyzed. By carefully studying the operational flow of vessel stowage, a multi-objective mixed integer programming model is put forward with regard to general principles. Then a specified genetic algorithm is proposed to solve the IP model. An integer encoding technique is employed in the algorithm, together with a self-crossover operator and a mutation operator. Furthermore, numerical tests are carried out and their results show the effectiveness and feasibility of the model. The application of the proposed theory provides a practical significance to improve loading efficiency and stowage quality.

## I. INTRODUCTION

With the ever-growing trend of economic and trade globalization, the majority of general cargo is nowadays containerized and there is an increasing requirement for mega containerships to be put into use in the maritime transportation system. Naturally then, container terminals secure a crucial position in the container transportation. Automated equipment in container terminal has elevated. Mi et al. (2013) has proposed a ship identification algorithm to identify cargo ships automatically. They (Mi et al., 2014) then proposed a fast

human-detection algorithm to supervise unmanned surveillance area in automated container terminal. A follow-up research (Mi et al., 2015) of human detection in automated container terminal has been significant. And We (Zhao and Shen, 2015) proposed a workflow engine based vehicle-mounted task control system modeling method to support process modeling. Based on these automation and modeling elevation and due to the intense competition among container terminals, the pressure of service quality improvement, service cost reduction and throughput increase occurs (He et al., 2010). Hence, the improvement of service level of the terminals has posed a challenging issue towards terminal operators.

For many container terminals, the operational process contains yard crane scheduling, quay crane scheduling, storage space allocation both at the quay side and yard side, berth allocation and so forth. Amongst, the stowage planning ensures great importance during the container loading operations. The problem addressed in this paper exactly refers to the position assignment for containers in a containership. It is a kind of loading problem, which means a detailed loading plan for pre-stowing containers of a specific vessel. In the late period of the last century, the vessel stowage was performed by the chief officer of a vessel. In contrast, the terminal today may decide the stowage plan in a more intelligent and reasonable manner with regard to the given instructions and constraints.

A containership is usually divided into multiple vessel bays. Each bay is split into the storage space on deck and in the hold, which is set apart by a hatch cover. Each location in the vessel is addressed by the following three identifiers: (a) bay, that gives its position relative to the cross section of the ship (counted from bow to stern); (b) stack, that gives its position relative to the vertical section of the vessel bay (counted from the center to the outside); (c) tier, that gives its position concerning the horizontal section of the bay (counted from the bottom to the top of the ship) (Ambrosino et al., 2004). During the loading period, each stack is assigned with a certain container group, which is identified by the port of destination and the container size. And so a vessel stack can be filled with any container as long as the container group planned for that stack is the same as or suitable for the required container group.

The process of stowage planning generally comprises of

---

Paper submitted 10/22/2015; revised 11/19/15; accepted 01/25/16. Author for correspondence: ZHAO Ning (email: ningzhao@shmtu.edu.cn).

<sup>1</sup>Logistics Engineering College, Shanghai Maritime University, 201306, Shanghai, China.

<sup>2</sup>Taicang Port SP Zhenghe Container Terminals Co. Ltd, Jiangsu Province, China.

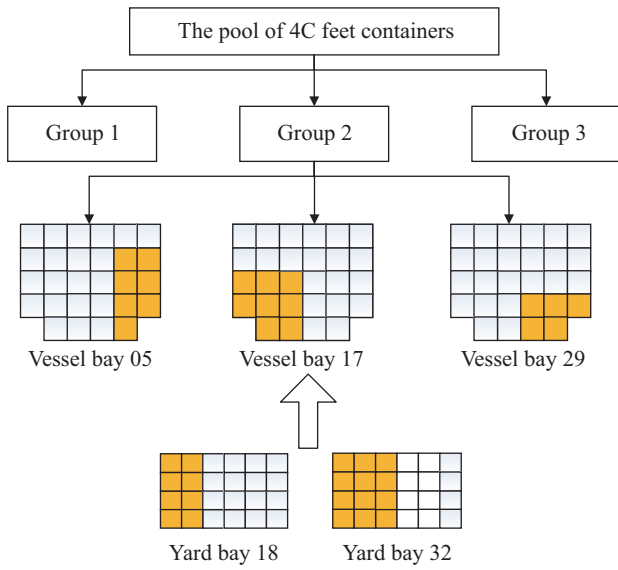


Fig. 1. A schematic diagram of stowage planning.

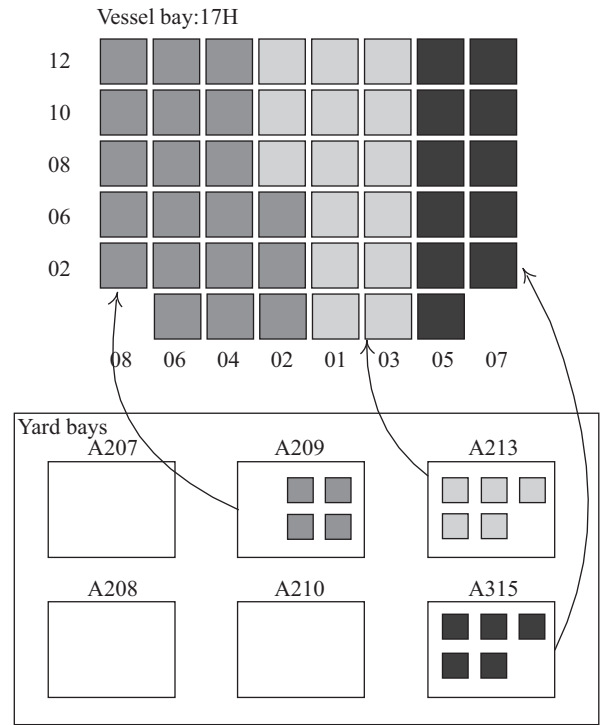


Fig. 3. A typical illustration of vessel slot planning.

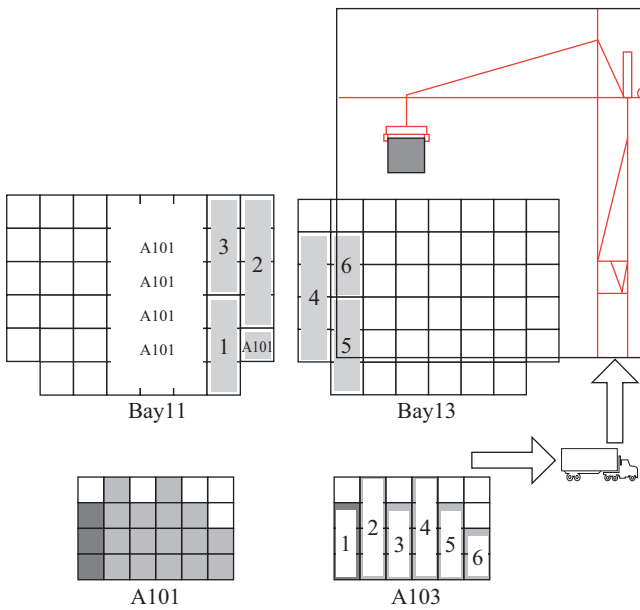


Fig. 2. The new concept of bay-filling.

five steps (see Fig. 1). Firstly, outbound containers of a vessel are classified into groups according to pre-stowing information provided by maritime companies, such as discharging ports, container sizes and shapes, container weight and so forth. In this regard, a complicated stowage process can be divided into several sections by container groups, which helps to simplify the problem. Secondly, for each container group, vessel bays are scheduled to provide a specific number of adjacent cells to hold the containers from the same group. In this way, the relations between container groups and vessel bays are formulated. Thirdly, the concept of bay-filling (Fig. 2)

is introduced. Bay-filling serves as an important link that fulfills the partitioning of adjacent cells in a vessel bay and the search for eligible containers in yard bays to match these cell groups. It can be interpreted as a phase during which containers from slots in yard bays are assigned to be retrieved and stowed into cell groups in a vessel bay according to the container distribution in the storage yard and pre-stowing plans provided by shipping corporations. This step is intended to manage the partitioning of yard operations and the movement of yard cranes. Fourthly, on the basis of the bay-filling result, a group of containers from yard bays are stowed into a single stack of the vessel bay. Accordingly, vessel slot planning is proposed (see Fig. 3). It can be composed of the following three procedures: 1) Select a target area in a vessel bay. The entire vessel bay may be partitioned into two or more sections due to the hatch cover in order to avoid the container repositioning in the storage yard. 2) Search and choose a container group to obtain the number of blocks and container distribution in the yard bays of each block. 3) Stow the selected containers to the stacks of a vessel bay. The detailed stowing sequence of each container is finally figured out and the cell allocation is established.

As shown in Fig. 1, the vessel contains three container groups and three vessel bays are planned to hold the containers from group 2. Totally there are 8 containers in bay 05, 7 containers in bay 17 and 5 containers in bay 29. And two yard bays are scheduled to release containers (8 containers in bay 18 and 12 containers in bay 32).

As a matter of fact, port operators as well as shipping

companies pay much attention to the vessel stowage planning and associated picking up or stacking operations, which may directly affect operational efficiency and terminal productivity in container terminals. On the one hand, the loading sequence made from stowage planning is very important for future container unloading operation. More specifically, the quality of a stowage plan is a crucial factor that determines ship's dwelling time in the port. The dwelling time of a ship includes the time for berthing, unloading, loading and departure, and therefore a smooth and orderly turnaround of a container vessel is essential for evaluating economic performance of liner shipping companies (Imai et al., 2006). On the other hand, the stowage plan must be made in accordance with the given pre-stowage plan and the restrictions enforced in retrieving containers from the stacks in storage yard. Each container can only be loaded into a cell in the hold or on the deck. The main disadvantage of the conventional operations lies in container rehandling, during which the additional unproductive moves have to be performed to retrieve a container from a lower tier where one or more containers are located over it. Container reshuffle is rather costly to the terminal operating company and it may be so serious as to lengthen the vessel's turnaround time and adversely affect the handling efficiency in a container terminal. And so, an orderly loading process should be guaranteed to effectively decrease container reshuffles. Meanwhile, the total number of unnecessary movement of quay cranes and yard cranes should be minimized as well.

In order to optimize the stowage process and overcome the afore-mentioned existing weakness in containerized shipping industry, the stowage planning problem for 40 feet containers is investigated in this paper. As such, it is instructive to explore an appropriate approach to solve the vessel stowage planning. As stated in the following sections, four important factors, based on the strict limit to the detailed rules and regulations in the Preliminary Stowage Plan (PSP), are carefully taken into account and they are outlined as follows: 1) Reshuffles of containers in storage yard; 2) Over-stowage in the containerships; 3) Idleness of quay cranes; 4) Remarshalling of yard cranes. On account of this, a state-of-the-art approach named 'ROIR', which covers very crucial aspects in the stowage process, is put forward and then employed. Even if the containers with different types are supposed to be stowed, the proposed problem is still applicable because the liner shipping company designates the vessel bay for each type of containers. Therefore, the suggested model is also feasible for all the other container types without further modification in the model and algorithm.

The rest of paper is structured as follows. In the next section, the literatures are reviewed. Section 3 explains the problem addressed in this paper and some key points in stowage planning are investigated, namely 'ROIR' strategy. In Section 4, the implementation of an integer programming model is put forward. And the genetic algorithm is proposed in Section 5. Then the detailed computational results are given

in Section 6. Conclusions are discussed in Section 7.

## II. LITERATURE REVIEW

Since 1970s, researchers all over the world have tried to examine and worked on the stowage planning problem. It can be widely recognized in the previous literatures from different points of view, using such methods as heuristics, simulation, design of decision support system, operations research and genetic algorithm, which have been gradually optimized in order to solve the problem more efficiently and accurately.

Shields (1984) developed a computer-aided containership stowage planning system, where only a small number of stowage plans were created and then evaluated by the simulation. Shortly after that, further investigations were carried out in (Ratcliffe and Sen, 1987; Saginaw and Parakis, 1989), they applied expert systems and rule-based techniques to assist container stevedoring to find out the suitable solutions. And furthermore, a rule-based decision system for dealing with master bay plan problem (MBPP) was presented by Ambrosino and Sciomachen (1998) for the first time. A constraint satisfaction approach was used to define and characterize the feasible solutions without employing an objective function to optimize the result. Winter et al. (1999) introduced stowage planning in connection with loading plans, taking the workload balance of quay cranes into consideration. These researches were intensively focused on ship stability. However, from our perspective, there is no need for port operators to think about stability in the stowage planning. Since it has been defined in the pre-stowing plans from shipping liners, some conditions have been confirmed and so it is not essential owing to the fact that containers from the same group can be stowed into a single vertical stack in the vessel. Containers in the same stack will be unloaded at the same destination. Excessive concerns about stability may exert great working pressure and unnecessary calculations on port operators.

For mathematical methods, Cho (1984) and Botter (1991) established the mathematical model and employed linear programming, which incorporated some hypotheses for the purpose of problem simplification. Nevertheless, it was not practical in the realistic process. Avriel and Penn (1993) and Avriel et al (1998) addressed a stowage problem, in which they formulated a 0-1 Integer Programming model and a heuristic called Suspensory Heuristic to stow the vessels. Ambrosino et al. (2004) addressed a stowage-planning problem with the objective to minimize the total stowage time where more practical constraints were taken into account such as different types of containers (in length) and weight limit accepted for securing ship structure. They assigned some ship holds to containers with the same destination in order to avoid unproductive unloading rehandles. Imai and Miki (1989) and Imai et al. (2001, 2002) carried out studies on loading operations at the container terminal. A multi-objective stowage planning model was established for a containership with container rehandles in the storage yard. They utilized the estimated

number of rehandles in order to think about the rehandles. Therein, container rehandle was estimated based on the expected number when retrieving each container as the first one to be taken in the block. In addition, the binary linear programming model for container stowage problem can be found in (Avriel and Penn, 1993; Flor, 1998). It was quite hard to find an optimal solution by using a binary model because of the large number of binary variables and the corresponding constraints. With regard to the special case, an optimal algorithm was developed. Avriel and Penn (1993) described a heuristics as Whole Column Heuristics Procedure. And subsequently, Avriel et al. (1998) proposed a different heuristic called Suspensory Heuristic Procedure, which was designed and tested on a large number of simulation cases. The quality of the result and the computation time were proved to be satisfying. However, this method could only manage a simplified problem. A main disadvantage was its inflexibility in dealing with the problem, where some of the assumptions were removed. Avriel et al. (2000) regarded the minimization of over-stowage as an NP-complete problem and they discovered heuristic methods to generate sound solutions. For this reason, a simulated annealing algorithm and a branch and bound algorithm were used to solve the shifting problem just as (Flor, 1998; Horn, 2000). Their success consisted in the flexibility in handling a variety of constraints that could be added to the basic problem. Unfortunately, only small sized problems could be solved by these heuristics. Additionally, the simulated annealing algorithm might lead to poor outcomes. Todd and Sen (1997) implemented a GA procedure with multiple criteria such as the proximity in terms of the container locations on board and minimization of unloading-related reshuffle. Their study also examined the relation between container reshuffles and the ship stability. This motivates us to take an attempt on genetic algorithm (GA). The genetic algorithm can handle the loading plans of a containership due to its parallel and non-linear nature of search. Moreover, it can manage a variety of constraints to be supplemented to the simplified problem.

Some researchers explored the potential of application of artificial intelligence. Wilson and Roach (1999, 2000), Wilson et al. (2001) presented a theoretical model, in which various technical restrictions were considered in order to realize the implementation of a commercial decision support system. Their approach was based on decomposing the planning process into two phases. In the first phase, called the strategic process, they made a rough stowage plan, based on classifying the containers with the same characteristics in terms of size, destination and etc. The calculations were performed by a branch and bound procedure. In the second phase, called the tactical process, individual containers were assigned to specific locations by using a tabu search heuristic, thus resulting in a detailed stowage plan. In addition to (Wilson and Roach, 1999; Wilson and Roach, 2000; Wilson et al., 2001), the slot planning optimization have been performed by a quite number of scholars over the past two decades. Some discussed about

the single phase planning model, and others divided it into multiple phases for generating plans. For the single phase, Avriel et al. (1998) considered all the containers with the same feature and tried to minimize container over-stowage. With Dubrovsky and Penn (2002), a genetic algorithm was shown to minimize the number of container movements. For the other one, Ambrosino et al. (2009, 2010) illustrated a tabu search heuristic to solve the same sub-problem and two new solution procedures were proposed, namely a fast simple constructive loading heuristic and an ant colony optimization algorithm. Kang et al. (2002) described an enumeration approach for solving a very simple vessel slot planning, where only over-stowage minimization and the classification of 400 containers after weighting were considered. Zhang et al. (2005) and Yoke et al. (2009) put forward multi-phase approaches where the problems solved during the slot planning phase were not independent of each other. Delgado et al. (2012) developed an approach that was able to generate near-optimal plans for large container vessels within a few minutes. The problem was decomposed into a master planning phase that distributed the containers to bay sections and a slot planning phase that assigned the containers of each bay section to slots. The majority of these papers only handled over-stowage problem.

Kim (1994, 1997), Kim and Kim (1994) and Kim et al. (2000) analyzed rehandles of transfer cranes and evaluated the number of rehandles. They (Kim et al. 2004) addressed a load planning problem with an objective of proper arrangement of container stacks on board in light of the smooth quay crane operation and the other one of proper container retrieval sequence from container stacks in the storage yard in view of an orderly transtainer operation, in which a beam-search algorithm was developed. More recently, Imai et al. (2006) tackled the problem to obtain a non-inferior solution for stowage problem. The problem was defined as a multi-objective integer programming, for which a set of non-inferior solutions was generated by using the weighting method. Sciomachen (2007) employed a 3D-BPP approach to optimize stowage plans and terminal productivity. They evaluated the performance of stowage plans so as to minimize the total loading time and ensure an effective use of quay cranes. However, in the process of on-site stowage planning, the evaluation of yard cranes and other factors are also decisive and they cannot be ignored. Lee and Lee (2010) presented a heuristic way for the optimization of a work plan, which was aimed to retrieve all the containers from a given yard according to a given order. The optimization goal was to minimize the number of container movements as well as the cranes' working time. A binary integer program was generated to reduce the length of the movement sequence and the sequence was iterated to shorten operational time. These researches mainly discussed container reshuffles in the storage yard.

As mentioned above, the studies on stowage planning have been extensively addressed. However, it can be noted that few research works have been carried out, which are dedicated to a

comprehensive investigation into the vessel stowage planning, especially the key elements described in the previous section. Hence, a novel ROIR approach is proposed and the problem is addressed on the basis of preliminary stowage plans. From an integrated viewpoint, this strategy helps to improve the operational efficiency at container terminals. It will be referred to as a practical, constructive and supplementary solution to the current research area of vessel stowage planning.

### III. PROBLEM DESCRIPTION

The purpose of timing synchronization is to allow the locally generated spreading signal to synchronize with the one embedded in the received signal. The timing synchronization is usually achieved in two stages: code acquisition and code tracking. The code acquisition is used to bring the timing offset between the received signal and the locally generated spreading signal to within the pull-in range of the code tracking loop, and then the code tracking can be initiated to correct the timing offset.

Stowage planning is an uppermost procedure when the outbound containers are planned to be loaded onto their target vessel. The planning is intended to assign each container with a specific location on the vessel where the stacking area has been specified by a preliminary stowage plan from the liner shipping company. But in special cases, the containers are not placed according to the plan and the chief officer has the right to refuse to sign a document for the ship's departure and the terminal will suffer from the penalty. As a result, the actual loading sequence must be made in order that each container is stowed into the right position.

The stowage planning of containers is closely related to the loading efficiency of a vessel. Once the stacking position in a containership is determined, some container reshuffles are inevitable and the yard cranes are required to move for an extra distance to perform the tasks. And furthermore, an improper stowage plan may lead to the potential efficiency decline of quay cranes. Some of the yard cranes (YCs) may interfere with each other without proper control. And even worse, the total number of all working YCs may be insufficient. Consequently, a sound stowage planning is definitely important.

Pertaining to the proposed model, a brief introduction of the stacking space both in the storage yard and in a vessel's hold is given in this section. Fig. 4 shows a block with 30 bays, 6 rows and 5 tiers, with a maximum capacity of 450 forty feet containers. Fig. 5 shows a vessel hold with 8 stacks, in which the bold line represents the hatch cover.

The following four sections will focus on ROIR, which has been presented in the first section. It is of practical significance on objectives and constraints in the model.

#### 1. Reshuffles of Containers in Storage Yard

The number of times that a container is reshuffled before the actual loading must be minimized owing that it may

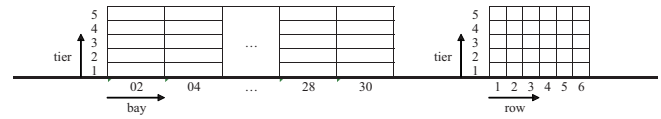


Fig. 4. An instance of one container block in the storage yard.

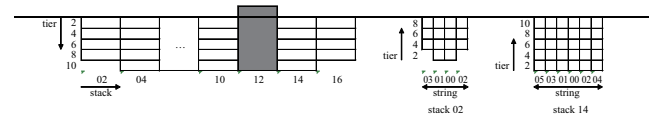


Fig. 5. A sketch of stacking locations in a vessel hold.

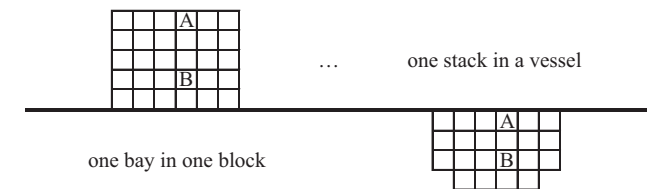


Fig. 6. A diagram showing the container reshuffle.

negatively affect the efficiency of picking up in the storage yard and increase the operational cost for the extra movements. Although the reduction in container rehandles will bring about large savings, it is impossible to completely eliminate the rehandling.

Generally, unnecessary reshuffles are caused by an unreasonable stowage plan. A typical example of the container reshuffle is shown in Fig. 6. Two containers indexed by A and B are stored in the same row of a yard bay, and herein the container A is located on the top of the row where the location is higher than that of container B. As to a specific planning, they are assigned to be stowed into the same stack of a vessel bay and similarly the container A is vertically higher than B. In this case, the container A has to be retrieved from the block earlier by a yard crane and temporarily placed somewhere else instead. It cannot be loaded until the container B has been put into the given location. However, the port operators don't hope so. Actually as a result, container reshuffles should be taken into consideration first and foremost and rationally controlled in the stowage plans.

#### 2. Over-Stowage in a Containership

Containers in a containership are stacked one on top of the other in stacks of a vessel bay, and can only be unloaded from the top of a stack. As described by Todd and Sen (1997), the classification of container weight should be observed. In other words, heavier containers should generally be placed at the lower layer than that of the other containers. The maximum allowable weight of a single stack should not be exceeded. During the process of stowage planning, the chief mate of a container vessel can reject the plan in case that the number of over-stowed containers is beyond permission.

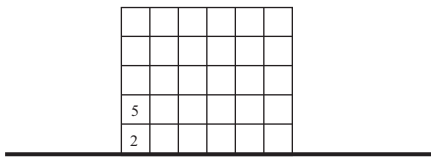


Fig. 7. Illustration of over-stowing containers in a vessel bay.

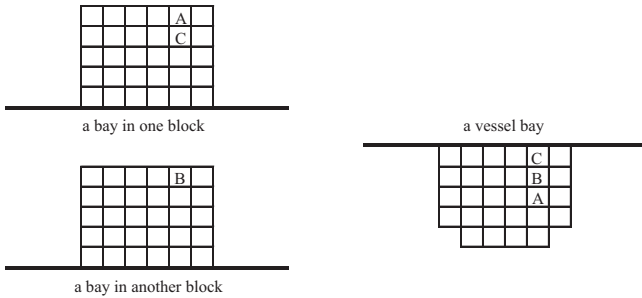


Fig. 8. A schematic of QC idleness.

In our study, a container is over-stowing another one in the same stack (see Fig. 7) if the heavier one is stowed above a lighter one or the order of two containers is carelessly reversed. To tell the truth, over-stowage is exactly expensive since the container must be removed by a quay crane to satisfy the corresponding requirements and principles.

3. Idleness of Quay Cranes

Diversity techniques, which are widely used for combating multipath fading effects, can be implemented in many ways. In this paper, we adopt a relatively simple yet effective spatial diversity technique called equal gain combining (EGC). The EGC combines the received signals from multiple hydrophones at different spatial locations to form a signal with a higher signal-to-noise ratio (SNR).

It is decided by the terminal operators that how yard cranes (YCs) and quay cranes (QCs) are combined to handle each container. There are some principles to judge whether the same YC and QC are deployed to load or unload a specific container, which are listed as follows. 1) Containers in the same bay or two adjacent bays in the same block are picked up by the same YC. 2) Containers assigned to the same stack or two stacks that are close to each other in a vessel bay are supposed to be handled by the same QC. 3) Containers from different blocks are retrieved by different YCs.

The handling efficiency of YCs is technically lower than that of QCs. Hence the containers to be loaded by the same QC are always retrieved by multiple YCs almost simultaneously. YC transfers between two blocks are always time-consuming and costly, which at the same time leads to the traffic congestion in the storage yard. Moreover, the interference between two or more YCs will further have a poor impact on the operations of QCs. QCs are required to wait until the target container comes.

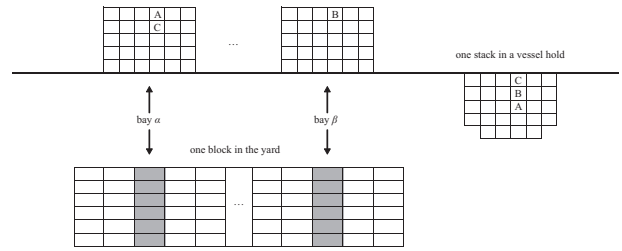


Fig. 9. An illustration of unnecessary YC remarshaling.

As shown in Fig. 8, container A and C are stacked in the same bay of one block while container B is in another neighboring block and they will be stowed into the same stack in a vessel. It is noted that all these containers are moved by the same YC. And therefore, the retrieval of B is delayed caused by the conflict between YC utilization. The quay crane has to wait for container B even if C has arrived at the quay-side, thus giving rise to a decrease in operational efficiency.

4. Remarshaling of Yard Cranes

Just like the example discussed in 3.3, the yard crane has to move back and forth to pick up all the containers (see Fig. 9). It is unwise to do so owing to the fact that the remarshaling of yard cranes brings about higher handling cost and longer operational time.

IV. MODEL FORMULATION

In this section, a multi-objective integer programming model is proposed, which is a representation of daily stowage planning for export containers.

1. Assumptions

The assumptions are listed as follows:

- (1) Only 40 feet containers are considered in the model. Moreover, reefer containers and dangerous containers are not taken into consideration.
- (2) The stacking position of each container in storage yard is known before the stowage plan is made.
- (3) Containers are only stowed in a vessel hold without containers on the deck.
- (4) There are enough cells in a containership and each container can be planned to any cell in a vessel bay, where the term ‘cell’ stands for the stowage position.

2. Notations

The parameters are provided as follows:

- $N$  The sum of containers stored in the storage yard, which is equal to the number of positions in a vessel’s hold
- $i, j$  Serial number of containers stored in storage yard.  $1 \leq i, j \leq C; i, j \in N^*$



$w_i$	The weight of container $i$
$NZ$	The sum of blocks in the storage yard
$z$	Serial number of blocks. $1 \leq z \leq NZ, z \in N^*$
$NB^z$	The sum of bays in block $z$
$b$	Serial number of yard bays. $\forall z, 2 \leq 2b \leq NB^z, b \in N^*$
$NR^z$	The number of rows in block $z$
$r$	Serial number of rows in bay $b$ . $\forall b, 1 \leq r \leq NR^z, r \in N^*$
$NT^z$	The number of tiers in block $z$
$t$	Serial number of tiers in bay $b$ . $\forall b, 1 \leq t \leq NT^z, t \in N^*$
$SY_{iz(2b)rt}$	$SY_{iz(2b)rt} = 1$ , if the container $i$ is stored in block $z$ , bay $b$ , row $r$ and tier $t$ ; $SY_{iz(2b)rt} = 0$ , otherwise
$D_{zbz'b}$	The cost for YC movement from bay $b$ of block $z$ to bay $b'$ of block $z'$
$m, n$	Serial number of assigned locations in a vessel's hold
$NC$	The number of vessel bays
$c$	Serial number of vessel bays. $2 \leq 2c \leq NC, c \in N^*$
$NS^{2c}$	The total number of stacks in the vessel bay $2c$
$s$	Serial number of all the stacks in a containership. $0 \leq s \leq NS^c, s \in N$
$NL^{(2c)s}$	The sum of tiers of the stack $s$ in vessel bay $2c$
$l$	Serial number of all the tiers in a vessel bay. $2 \leq 2l \leq NL^{(2c)s}, l \in N^*$
$R^{2c}$	A recommended value for the number of blocks, where the container to be stowed into the vessel bay $2c$ is stacked in row $r$
$SV_{m(2c)s(2l)}$	$SV_{m(2c)s(2l)} = 1$ if the cell $m$ in a vessel bay can be expressed by vessel bay $2c$ , stack $s$ and tier $l$ ; $SV_{m(2c)s(2l)} = 0$ , otherwise
$NG$	The total number of container groups, which is equal to the number of stacks in a specific vessel bay
$g$	Serial number of all the groups. $1 \leq g \leq NG, g \in N^*$

The variables are listed as follows:

$x_{im}$	$x_{im} = 1$ , if the container $i$ is assigned to the cell $m$ ; $x_{im} = 0$ , otherwise
$ow_{ij}$	$ow_{ij} = 1$ , if the weight of container $i$ is heavier than $j$ ; $ow_{ij} = 0$ , otherwise
$oz_{ij}$	$oz_{ij} = 1$ , if container $i$ is higher than $j$ ; $oz_{ij} = 0$ , otherwise
$ov_{mn}$	$ov_{mn} = 1$ , if the cell $m$ is multiple tiers higher than $n$ , where they are in the same stack of a vessel bay; $ov_{mn} = 0$ , otherwise
$ot_{mn}$	$ot_{mn} = 1$ , if the cell $m$ is only one tier higher than $n$ , where they are in the same stack of a vessel bay; $ot_{mn} = 0$ , otherwise
$nd_{ij}$	The movement cost for QCs from the vessel bay of container $i$ to that of container $j$
$dc_{(2c)sz}$	The total number of containers that is stored in

	block $z$ and to be placed in the stack $s$ of the vessel bay $2c$
$om_{(2c)sz}$	$om_{(2c)sz} = 1$ , if block $z$ is the target block for the stack $s$ in the vessel bay $2c$ ; $om_{(2c)sz} = 0$ , otherwise
$dd_{(2c)}$	The maximum difference between the number of containers from some target blocks

### 3. Objectives and Constraints

As mentioned in the previous sections, there are mainly four sub-problems that should be taken into account when making a stowage plan for the outbound containers of a vessel. The first sub-problem is the minimization of unavoidable reshuffles of yard cranes. The second one is the minimization of over-stowing containers. The third one is the minimization of the probability of QC idleness caused by YC conflict. The last one is the minimization of unnecessary movement of yard cranes. These problems are formulated in the following three objectives.

#### 1) Minimization of Unavoidable Reshuffles

During the process of container loading operations, unnecessary container reshuffles will certainly require the extra workloads of yard cranes. This may further add to the burdens of the vessel handling operation. Therefore, it is treated as the most important objective in this paper, and can be described as follows.

$$\text{Min } f_{ur} = \text{Min} \sum_{i=1}^N \sum_{j=1}^N \sum_{m=1}^N \sum_{n=1}^N (oz_{ij} \cdot ov_{mn} \cdot x_{im} \cdot x_{jn}) \quad (1)$$

$$\sum_{i=1}^N \sum_{z=1}^{NZ} \sum_{b=1}^{NB^z/2} \sum_{r=1}^{NR^z} \sum_{t=1}^{NT^z} (z \cdot SY_{iz(2b)rt}) = \sum_{j=1}^N \sum_{z=1}^{NZ} \sum_{b=1}^{NB^z/2} \sum_{r=1}^{NR^z} \sum_{t=1}^{NT^z} (z \cdot SY_{jz(2b)rt}) \quad (2)$$

$$\sum_{i=1}^N \sum_{z=1}^{NZ} \sum_{b=1}^{NB^z/2} \sum_{r=1}^{NR^z} \sum_{t=1}^{NT^z} (b \cdot SY_{iz(2b)rt}) = \sum_{j=1}^N \sum_{z=1}^{NZ} \sum_{b=1}^{NB^z/2} \sum_{r=1}^{NR^z} \sum_{t=1}^{NT^z} (b \cdot SY_{jz(2b)rt}) \quad (3)$$

$$\sum_{i=1}^N \sum_{z=1}^{NZ} \sum_{b=1}^{NB^z/2} \sum_{r=1}^{NR^z} \sum_{t=1}^{NT^z} (r \cdot SY_{iz(2b)rt}) = \sum_{j=1}^N \sum_{z=1}^{NZ} \sum_{b=1}^{NB^z/2} \sum_{r=1}^{NR^z} \sum_{t=1}^{NT^z} (r \cdot SY_{jz(2b)rt}) \quad (4)$$

$$\sum_{i=1}^N \sum_{z=1}^{NZ} \sum_{b=1}^{NB^z/2} \sum_{r=1}^{NR^z} \sum_{t=1}^{NT^z} (t \cdot SY_{iz(2b)rt}) - \sum_{j=1}^N \sum_{z=1}^{NZ} \sum_{b=1}^{NB^z/2} \sum_{r=1}^{NR^z} \sum_{t=1}^{NT^z} (t \cdot SY_{jz(2b)rt}) > 0 \quad (5)$$

$$oz_{ij} = \begin{cases} 1, & \text{when the equations (2-5) are satisfied} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

$$\sum_{m=1}^N \sum_{c=1}^{NC/2} \sum_{s=0}^{NS^{2c}} \sum_{l=1}^{NL^{(2c)s}/2} (c \cdot SV_{m(2c)s(2l)}) = \sum_{n=1}^N \sum_{c=1}^{NC/2} \sum_{s=0}^{NS^{2c}} \sum_{l=1}^{NL^{(2c)s}/2} (c \cdot SV_{n(2c)s(2l)}) \quad (7)$$

$$\sum_{m=1}^N \sum_{c=1}^{NC/2} \sum_{s=0}^{NS^{2c}} \sum_{l=1}^{NL^{(2c)s}/2} (s \cdot SV_{m(2c)s(2l)}) = \sum_{n=1}^N \sum_{c=1}^{NC/2} \sum_{s=0}^{NS^{2c}} \sum_{l=1}^{NL^{(2c)s}/2} (s \cdot SV_{n(2c)s(2l)}) \quad (8)$$

$$\sum_{m=1}^N \sum_{c=1}^{NC/2} \sum_{s=0}^{NS^{2c}} \sum_{l=1}^{NL^{(2c)s}/2} (l \cdot SV_{m(2c)s(2l)}) - \sum_{n=1}^N \sum_{c=1}^{NC/2} \sum_{s=0}^{NS^{2c}} \sum_{l=1}^{NL^{(2c)s}/2} (l \cdot SV_{n(2c)s(2l)}) < 0 \quad (9)$$

$$ov_{mn} = \begin{cases} 1, & \text{when the equations (7-9) are satisfied} \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

### 2) Minimization of Over-Stowing Containers

The number of over-stowing containers has been the most important factor to consider in the stowage planning for a period of time when the container vessels were not large enough and ship stability could be affected easily with a few over-stowing containers, whereas the vessels today are generally larger than before. For this reason, it is regarded as the second objective and can be expressed as follows.

$$\text{Min } f_{th} = \text{Min} \sum_{i=1}^N (ow_{ij} \cdot ot_{mn} \cdot x_{im} \cdot x_{jn}) \quad (11)$$

$$ow_{ij} = \begin{cases} 1, & \text{when } w_i > w_j \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

$$\sum_{n=1}^N \sum_{c=1}^{NC/2} \sum_{s=0}^{NS^{2c}} \sum_{l=1}^{NL^{(2c)s}/2} (n \cdot SV_{m(2c)s(2l)}) - \sum_{m=1}^N \sum_{c=1}^{NC/2} \sum_{s=0}^{NS^{2c}} \sum_{l=1}^{NL^{(2c)s}/2} (m \cdot SV_{n(2c)s(2l)}) = 2 \quad (13)$$

$$ot_{mn} = \begin{cases} 1, & \text{when equations (7-8) and (13) are satisfied} \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

### 3) Minimization of QC Idleness and Unavoidable YC Re-marshalling

The cause of potential QC idleness is similar to that of the unnecessary YC movement. And so these two sub-problems can be integrated into one objective, which is objective could be stated as follows.

$$\text{Min } f_{qw} = \text{Min} \sum_{i=1}^N \sum_{j=1}^N \sum_{m=1}^N \sum_{n=1}^N (nd_{ij} \cdot ot_{mn} \cdot x_{im} \cdot x_{jn}) \quad (15)$$

$$nd_{ij} = \sum_{z=1}^{NZ} \sum_{b=1}^{NB^z/2} \sum_{z'=1}^{NZ} \sum_{b'=1}^{NB^{z'}/2} \left( D_{z(2b)z'(2b')} \cdot \sum_{r=1}^{NR^z} \sum_{t=1}^{NT^z} (SY_{iz(2b)rt} \cdot SY_{jz'(2b')rt}) \right) \quad (16)$$

### 4) Other Constraints

It must be noted that it is a one-to-one relation between containers and cells in a vessel bay. These constraints are expressed in the following equations.

$$\forall i, \sum_{z=1}^{NZ} \sum_{b=1}^{NB^z/2} \sum_{r=1}^{NR^z} \sum_{t=1}^{NT^z} SY_{i(2b)rt} = 1 \quad (17)$$

$$\forall m, \sum_{c=1}^{NC/2} \sum_{s=0}^{NS^{2c}} \sum_{l=1}^{NL^{(2c)s}} SV_{m(2c)s(2l)} = 1 \quad (18)$$

$$\forall i, \sum_{m=1}^N x_{im} = 1 \quad (19)$$

$$\forall m, \sum_{i=1}^N x_{im} = 1 \quad (20)$$

Constraint (17) defines the locations of containers in the storage yard. And the positions in a vessel's hold are determined in constraint (18). Constraint (19) ensures that one container is planned to be placed into only one cell in a vessel. Constraint (20) means that one cell in a vessel's hold is reserved for only one container.

## V. SOLUTION METHOD

A mixed integer programming model is specified in the last section, which describes multiple objectives with various priorities. However, such transportation systems as container terminals are too complicated to be solved analytically. Additionally, despite that the methods like simulation or operations research can obtain a final result and evaluate the operational and economic performance, it is not well-desired. And consequently, an optimization algorithm is needed to enhance the computational quality.

The algorithm is intended to classify all the containers properly into groups, in which each group is connected with a certain stack in a vessel's hold with the same loading capacity. The order of container groups makes no difference to the stowage plan in view that various orders enjoys the same fitness value. As a result, a well-designed genetic algorithm will be enough to solve the model. The term 'candidate' is raised from the perspective of the problem analysis, which is similar to 'Chromosome'.

The following key points are employed in the proposed algorithm:

- An integer coding method with both a self-crossover and a self-mutation operator, which helps to keep the uniqueness of containers in each candidate.
- A sorting strategy that leads to the best container sequence in a container group.
- A self-crossover operator with allowed crossovers among container groups.
- A mutation operator that leads to a better candidate if possible.
- An iteration mechanism that utilizes operators.

### 1. Notations of the Algorithm

The notations are listed as follows.

- $NG$  The maximum number of generations
- $g$  Serial number of generations.  $1 \leq g \leq NG, g \in N^*$
- $NS$  Population scale
- $s$  Serial number of candidates.  $1 \leq s \leq NS, s \in N^*$
- $NP$  Number of container groups in each candidate
- $p$  Serial number of container groups in a candidate.  $1 \leq p \leq NP, p \in N^*$
- $NC_p$  Number of containers in group  $p$
- $q$  Serial number of containers in group  $p$ .  $1 \leq q \leq NC_p, q \in N^*$
- $R_q^{gsp}$  The movement cost from the container  $q$  to container  $q + 1$  in group  $p$  of candidate  $s$  in generation  $g$
- $oh_q^{gsp}$   $oh_q^{gsp} = 1$ , if the container  $q$  is heavier than container  $q + 1$  in group  $p$  of candidate  $s$  in generation  $g$ .  $oh_q^{gsp} = 0$ , otherwise
- $DM$  The upper limit of  $D_{zbz'b'}$
- $dx_{rsp}$  The number related to the possibility if group  $p$  is chosen for the crossover in candidate  $s$  of generation  $g$
- $rx_{gsp}$  The probability value if group  $p$  is chosen for the crossover in candidate  $s$  of generation  $g$ .  $0 \leq rx_{gsp} \leq 1$
- $B_{abgs}$  The benefit from the exchange between gene  $a$  and  $b$  of candidate  $s$  in generation  $g$  during the mutation process.  $1 \leq a, b \leq \sum_{p=1}^{NP} NC_p$
- $NX$  Number of candidates chosen for crossover in each generation
- $rx_{gs}$  The possibility of the crossover for candidate  $s$  in generation  $g$
- $rxp_{gsp}$  The possibility of the crossover for group  $p$  of candidate  $s$  in generation  $g$
- $f_{gsp}$  Fitness value of group  $p$  of candidate  $s$  in generation  $g$
- $f_{gs}$  Fitness value of candidate  $s$  in generation  $g$
- $nk_g$  The number of candidates remained by selection operator in generation  $g$  that is generated just in that generation
- $nn$  The number of successive generations in which no candidates created in current generation will be remained by the selection operator
- $NN$  The maximum allowable value for  $nn$

### 2. Encoding of Candidates

One candidate is encoded as multiple container groups. The number of groups ( $NP$ ) is equivalent to the number of stacks in a vessel bay, and the number of containers in each group ( $NC_p$ ) equals to capacity of the corresponding stack. The parameters  $NP$  and  $NC_p$  are defined as the following equations.

$$NP = \sum_{c=1}^{NC/2} NS^{2c} \quad (21)$$

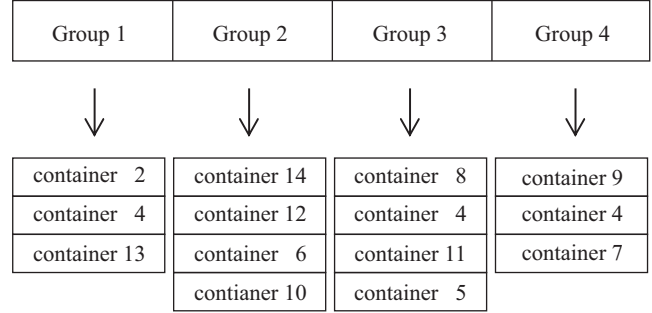


Fig. 10. Example of encoding.

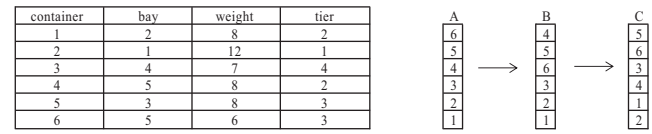


Fig. 11. An outline of sorting a container group.

$$NC_p = NL^{(2c)s}, p = \sum_{a=1}^{c-1} NS^{2a} + s - 1 \quad (22)$$

A typical example is illustrated in Fig. 10, where 14 containers are coded into four container groups. Two of them contains 3 containers for each while there are four containers in the others.

### 3. Initialization

The process of initialization consists of two phases. In the first phase, various random integer sequences with the length of  $N$  are continuously produced to be candidates in the first generation until the number of them reaches  $NS$ . These candidates are composed of containers with serial numbers, which appear only once in every candidate. In the second phase, the containers in each integer sequence are divided into groups according to  $NP$  and  $NC_p$ , and the fitness value of each candidate is computed as well. Compared with earlier candidates, the new candidate with the same containers in every group will be eliminated.

### 4. A Sorting Strategy for Container Groups

The fitness value of a container group depends on both the containers from this group and the stacking sequence of these containers. The recorded sequence of containers in a certain group should be justified before determining the group's fitness value. With reference to the objective functions (1, 11, 15), a multi-phase sorting strategy is put forward for containers in the same group, which is described as follows and exemplified in Fig. 11.

Step 1: Sort containers in accordance with container number (no. for short), where a unique sequence of the containers is formulated.

Step 2: Adjust the order of containers, which originally brings about unavoidable reshuffles. Then exchange the order among containers that are stacked in the same bay and same row in one block. It won't stop until there exist no container reshuffles.

Step 3: Sort containers using some sorting techniques. Two adjacent containers can exchange their positions as long as this exchange leads to no unnecessary reshuffle and also satisfies one of the following situations.

- The container in the upper position is heavier than the other one beneath it.
- Two containers almost have the same weight and the exchange can save movement cost.

As shown in Fig. 11, 6 containers are to be stowed into a specific stack in a vessel bay. The weight of containers and the stacking parameters for bays and tiers are listed in the left part of the figure. For step 1, these containers are arranged by container no. as shown in column A. As to step 2, the positions of container 4 and 6 are interchanged which is presented in column B, otherwise there will be an unavoidable reshuffle. And column C shows the final sequence of containers in this group, which is determined by using Bubble Sorting Technique (BST) pertaining to the rules stated in step 3.

## 5. Fitness Evaluation

On the basis of a sorted container group of a candidate, it is quite easy to define the fitness function, and these fitness values are later added to obtain the total fitness value of the candidate. With regard to the equations (11) and (15) respectively, the fitness functions for group  $p$  of candidate  $s$  in generation  $g$  are expressed as follows.

$$f_{gsp} = \sum_{q=1}^{NC_p-1} (DM \cdot o h_q^{gsp} + R_q^{gsp}) \quad (23)$$

$$f_{gs} = \sum_{p=1}^{NP} f_{gsp} \quad (24)$$

## 6. Self-Crossover Operator

In standard Genetic Algorithm, crossover operator is employed to generate new chromosomes by exchanging two gene segments with the same length between current chromosomes. However, this crossover mechanism is not appropriate for the fact that every gene should be unique. The crossover between two candidates will easily break the uniqueness of container no. in every candidate. And so, a self-crossover operator is put forward in this paper. It only interchanges containers from two different groups of the same candidate, thus avoiding infeasible candidates.

During the process of iteration, a number of candidates ( $NXG$ ) in the current population are randomly selected for crossover. For a candidate with a more attractive fitness value,

it is more likely to be chosen. Once a candidate is determined to carry out crossover, self-crossover operator reallocates the containers in two groups of the candidate in order to produce a new candidate. The operator works in the following three steps.

Step1: Choose two groups from a candidate randomly and withdraw all containers. The crossover probability of a group ( $rx_{gsp}$ ) is calculated as the following equations.

$$rx_{gsp} = dx_{gsp} / \sum_{p=1}^{NP} dx_{gsp} \quad (25)$$

$$dx_{gsp} = \left( f_{gsp} - \text{Min}_p(f_{gsp'}) \right) / \left( \text{Max}_p(f_{gsp'}) - \text{Min}_p(f_{gsp'}) \right), 1 \leq p' \leq NP \quad (26)$$

Step 2: Select one container for each group as an initial container. For each container, the probability of selection is much related to the summation of distance between itself and another container in the group. The probability will be larger if the total distance is shorter.

Step 3: Allocate the rest of containers to the groups one after another. It is the fitness value that decides whether a container is assigned to a group. With regard to the group without any container, the container is more probable to be allocated if the fitness value of the group is lower. Once the allocation of all the containers terminates, a new candidate is generated and will be added to the population.

## 7. Mutation Operator

Similar to the crossover operator, standard mutation operations will have trouble in solving the problem. Once the container number of a certain gene changes, it should be a must that the container number of another gene in the same candidate changes as well. Otherwise, there will be two genes with the same container no. in a candidate, which violates the uniqueness of the container number. As a result, the proposed mutation operator is designed to exchange the container number between two genes in the same candidate.

It is determined that the total number of potential exchanges among genes in a candidate is limited. Hence, it is required to evaluate the performance of each possible exchange during mutation operations. Suppose that the mutation is triggered between container  $a$  in group  $A$  and container  $b$  in group  $B$ , and two groups are denoted by group  $A'$  and  $B'$ , which helps to facilitate the calculation of the effect of current operation. The expression can be defined as follow.

$$B_{abgs} = f_A + f_B - f_{A'} - f_{B'} \quad (27)$$

Mutation operations are carried out by chance in the standard genetic algorithm. It is rather hard to predict the consequence of each mutation and whether a mutation can generate a better candidate. However, the effect of every gene ex-

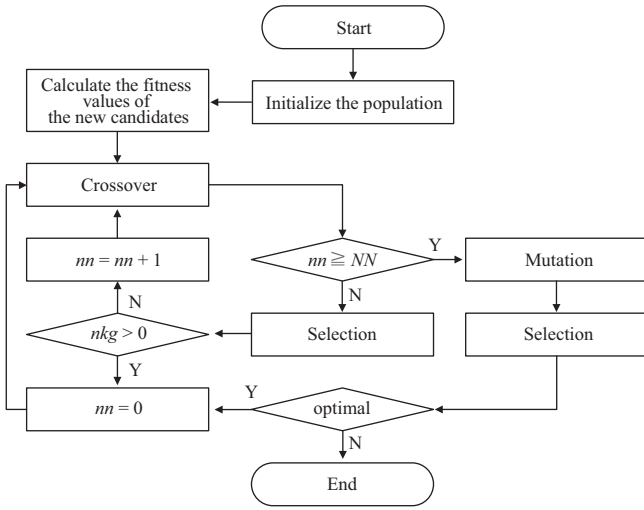


Fig. 12. An overall flowchart of a specified genetic algorithm.

change is always predictable in an integer coded candidate. In this case, the mutation operator is designed as an operator and the best mutation plan can be created only when the algorithm method converges to a local solution rather than by chance in every iteration. On the one hand, the utilization of a mutation operator should be limited during the searching process of the algorithm considering the large calculation cost. On the other hand, however, the operator can be further applied to evaluate whether the current procedure is terminated or not.

### 8. Selection Operator

As a method employed to keep the scale of population during the searching process, selection operator removes redundant candidates in every generation and generates a new sequence for current candidates by the fitness value. Herein, only the leading  $NS$  candidates are kept for the next generation.

### 9. Iteration Mechanism and Termination Criterion

The iteration mechanism and termination criterion can be defined as follows.

Step1: Only the crossover and selection operations are executed in every generation. Let  $nn = nn + 1$  if no candidate generated in the current generation are kept by the selection operator ( $nkg$ ); otherwise, let  $nn = 0$ .

Step2: If  $nn$  reaches  $NN$ , the mutation operator will be executed after the crossover operator. Let  $nn = 0$  if a more optimal value is generated by mutation operation and go back to step 1; otherwise, terminate the entire process.

In summary, the overall structure of the algorithm is shown in Fig. 12.

## VI. NUMERICAL EXPERIMENTS

A specified Genetic Algorithm for stowage planning model has been defined in the last section. Nevertheless, there are

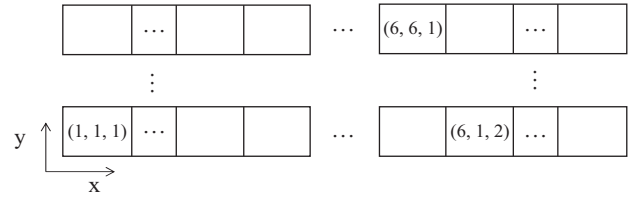


Fig. 13. An illustration of three-dimensional coordinates in the storage yard.

still two questions to be dealt with. The first question is how to obtain the most feasible performance of the algorithm with such parameters as the population scale ( $NS$ ), the number of candidates that are crossed in a generation ( $NX$ ) and the maximum allowable number of generations where no new candidate produced is kept by the current generation ( $NN$ ). The second question is what the performance of the algorithm will be if required to solve a large-scale problem.

To validate and verify the effectiveness and reliability of the proposed ROIR strategy and solution method as aforementioned, numerical tests with different sizes are conducted, which is aimed at addressing the issues with small sizes and large sizes. All instances are run on a PC with Intel Core (TM) i5 2520 M CPU @ 2.5 GHz processor and 3.2 GB RAM.

### 1. Case Description

The algorithm is tested on two practical instances from a container terminal in China. Both of the cases are only composed of 40 feet containers, 129 containers and 293 containers respectively. These two instances are actually large-size instances in today's container terminal owing that the number of containers to be arranged in a stowage plan is usually less than 100. For those with more than 100 containers, it will be appreciated if a stowage plan can be made and accomplished in no more than an hour.

The stacking positions of containers in the storage yard (see Fig. 13) are all known before the actual stowage plan. The blocks in the storage yard can be regarded as a matrix array and the position of each block is defined by a pair of  $x$ - $y$  coordinates. Hence, the movement cost of YCs can be computed in terms of three dimensions.

The notations used in defining the movement cost are listed as follows.

- $xz_z$  The  $x$ -axis of a block with stacking position  $z$
- $yz_z$  The  $y$ -axis of a block with stacking position  $z$
- $\lambda_x$  A coefficient in  $x$  dimension
- $\lambda_y$  A coefficient in  $y$  dimension

Consequently, the movement cost of YCs is defined in the following equation. The parameters are expressed by  $\lambda_x = 100$  and  $\lambda_y = 500$  respectively in this paper.

$$D_{z_bz'_b'} = \left| \lambda_x(x_z - x_{z'}) + \lambda_y(y_z - y_{z'}) + (b_z - b_{z'}) \right| \quad (28)$$

**Table 1. The average optimal fitness value in various pairs of NS and NX.**

<i>NX/NS</i>	5	10	15	20
5	683.2	779.6	505.6	609.2
10	634.0	621.6	532.3	543.6
15	578.8	576.4	477.2	456.1
20	620.4	534.8	500.8	507.2
25	--	575.6	462.8	458.8
30	--	554.3	402.4	364.0
35	--	--	443.1	374.4
40	--	--	--	436.4
45	--	--	--	406.7
50	--	--	--	--

**Table 2. The average solution time in various pairs of NS and NX.**

<i>NX/NS</i>	5	10	15	20
5	02:36.4	05:58.0	07:34.8	13:05.3
10	02:14.2	03:53.3	06:54.7	10:05.3
15	02:38.5	05:13.3	06:53.7	07:45.0
20	02:47.0	03:38.8	05:54.6	06:34.0
25	--	03:45.4	04:25.1	05:31.0
30	--	--	03:41.7	05:37.7
35	--	--	04:21.1	04:40.9
40	--	--	--	05:10.3
45	--	--	--	06:43.7
50	--	--	--	--

## 2. Numerical Tests with Small Sizes

To find the relations among *NS*, *NX* and *NN*, the small-size instances are operated for many times. *NN* is firstly determined to reduce the required solution time. After the relationship between *NX* and *NS* is discovered, more experiments are carried out to analyze three parameters.

### 1) Relationship Between NS and NX

Let *NN* be 10. For each pair of *NS* and *NX*, the real case is run for 20 times and the average optimal fitness value and solution time is recorded in Table 1 and Table 2 respectively.

It is found from table 1 and 2 that:

- The group with a larger *NS* is more likely to find a better solution on the condition that the value of *NX* is properly chosen. It should be noted that a longer time is required to search a feasible solution.
- With regard to *NS*, a number for *NX*, namely  $NX_{best}$ , can help to produce the best solution. Meanwhile, the solution time is likely to be the shortest. This can be described as the following expression.

**Table 3. The average optimal fitness value in various pairs of NS and NN.**

<i>NN/NS</i>	5	10	15	20
<i>NS</i>	749.5	410.2	495.9	810.1
2 <i>NS</i>	539.6	548.0	369.0	428.5
3 <i>NS</i>	517.2	554.6	352.6	404.7
4 <i>NS</i>	507.6	424.8	476.7	324.6
5 <i>NS</i>	634.8	508.7	427.3	398.4

**Table 4. The average solution time in various pairs of NS and NN.**

<i>NN/NS</i>	5	10	15	20
<i>NS</i>	03:39.3	04:52.6	06:17.5	05:55.4
2 <i>NS</i>	02:57.9	02:57.3	05:29.0	06:18.5
3 <i>NS</i>	03:06.2	04:48.3	06:00.0	06:11.0
4 <i>NS</i>	02:28.1	03:44.2	03:49.3	05:53.0
5 <i>NS</i>	02:20.1	04:16.5	03:16.2	03:37.0

**Table 5. Result for large size instances.**

<i>NS</i>	Avg. optimal fitness value	Avg. solution time
5	652.4	28:28.4
10	580.3	34:06.1
15	556.9	40:44.2
20	516.7	48:37.5

$$NX_{best} = 2NS \quad (29)$$

- For those with nearly 100 containers, it is suggested that  $NS = 20$  so that the result can be generated in 5 minutes.

### 2) Relationship Between NS and NN

Let  $NX = 2NS$ . Experiments are conducted to find out the relationship between *NN* and *NS*.

Similar to Tables 1 and 2, it can also be observed from Tables 3 and 4 that larger *NS* leads to a better optimal solution and the solution time is rather time-consuming. However, the relationship between *NS* and *NN* is not that obvious. It is drawn that a better optimal solution in a comparatively short time period will be generated when  $NN = 4NS$ . At the same time a larger *NN* will not trigger a better solution. As a result, it is recommended that the value of *NN* is set four times the value of *NS*.

## 3. Numerical Tests with Large Sizes

To further certify the effectiveness of the algorithm in solving large-size instances for stowage planning, the experiments are carried out to solve a planning problem including 293 containers. The population scales are 5, 10, 15 and 20 respectively. For each population, it is run for 5 times. Both

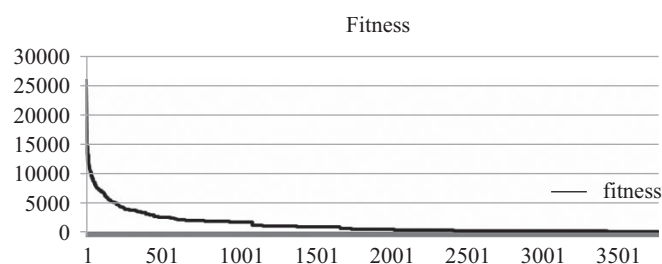


Fig. 14. Convergence curve for large-size instances.

the average optimal fitness value and solution time are listed in the following table. It can be concluded that the proposed algorithm is able to obtain a solution in no more than 50 minutes, which satisfies the practical needs.

As shown in Fig. 14, the curve conforms to a standard convergence, which is a strong evidence for presenting the effectiveness and reliability of the propose algorithm.

## VII. CONCLUSION

In this paper, we discuss the stowage planning problem for containerhips, in which several key principles are considered and analyzed. The problem is illustrated by a multi-objective integer programming model, which is a specific model covering important aspects in the vessel stowage. Based on this, a specified genetic algorithm is proposed to solve the model. It is used in the algorithm an integer encoding technique with both a self-crossover operator and an exchange-based mutation operator, which helps to keep the uniqueness of the container number in the iteration process. Afterwards, numerical experiments illustrate that this algorithm can always generate a good solution within a time period that satisfies the practical demand. However, the algorithm spends relatively long time in solving the instances with a very large problem scale (around 200 containers). Future researches are required to improve the performance of the algorithm.

## ACKNOWLEDGEMENTS

This research was supported by “Scientific Research Innovation Project” of Shanghai Municipal Education Commission (No. 14ZZ140), the “Ph. D. Innovation Program” of Shanghai Maritime University (No. 2014ycx040), and the “Yang Fan Plan for Shanghai Youth Science and Technology Talent” of the Science and Technology Commission of Shanghai Municipality (No. 15YF1404900).

## REFERENCES

- Ambrosino, D., D. Anghinolfi, M. Paolucci and A. Sciomachen (2009). A new three-step heuristic for the master bay plan problem. *Maritime Economics and Logistics* 11(1), 98-120.
- Ambrosino, D., D. Anghinolfi, M. Paolucci and A. Sciomachen (2010). An experimental comparison of different heuristics for the master bay plan problem. *Proceedings of the 9<sup>th</sup> International Symposium on Experimental Algorithms*, 314-325.
- Ambrosino, D. and A. Sciomachen (1998). A constraints satisfaction approach for master bay plans. In: Sciutto, G., Brebbia, C. (Eds.), *Maritime Engineering and Ports* 39, 155-164.
- Ambrosino, D., A. Sciomachen and E. Tanfani (2004). Stowing a container-ship: The master bay plan problem. *Transportation Research A* 38(2), 81-99.
- Avriel, M. and M. Penn (1993). Exact and approximate solutions of the container ship stowage problem. *Computers & Industrial Engineering* 25, 271-274.
- Avriel, M., M. Penn, N. Shpirer and S. Witteboon (1998). Stowage planning for container ships to reduce the number of shifts. *Annals of Operations Research* 76, 55-71.
- Avriel, M., M. Penn and N. Shpirer (2000). Container ship stowage problem: Complexity and connection to the coloring of circle graphs. *Discrete Applied Mathematics* 103, 271-279.
- Botter, R. C. and M. A. Brinati (1991). Stowage container planning: a model for getting an optimal solution. *IFIP Trans. B (App. in Tech.) B-5* 217-229.
- Cho, D. W. (1984). Development of a methodology for containerhip load planning. Ph. D. Thesis, Oregon State University.
- Delgado, A., R. M. Jensen and K. Janstrup, T. H. Rose and K. H. Andersen (2012). A constraint programming model for fast optimal stowage of container vessel bays. *European Journal of Operational Research* 220, 251-261.
- Dubrovsky, O., G. Levitin and M. Penn (2002). A genetic algorithm with a compact solution encoding for the container ship stowage problem. *Journal of Heuristics* 8, 585-599.
- Flor, M. (1998). Heuristic Algorithms for solving the container ship stowage problem. M. Sc. Thesis, Faculty of Industrial Engineering and Management, Technion, Haifa, Israel. (in Hebrew)
- He, J. L., D. F. Chang, W. J. Mi and W. Yan (2010). A hybrid parallel genetic algorithm for yard crane scheduling, *Transportation Research Part E* 46(1), 136-155.
- Horn, G. P. (2000). A branch and bound algorithm for the container ship stowage problem. M. Sc. Thesis, Faculty of Industrial Engineering and Management, Technion, Haifa, Israel. (in Hebrew)
- Imai, A. and T. Miki (1989). A heuristic algorithm with expected utility for an optimal sequence of loading containers into a containerized ship. *Journal of Japan Institute of Navigation* 80, 117-124. (in Japanese)
- Imai, A., E. Nishimura, S. Papadimitriou and K. Sasaki (2002). The containerhip loading problem. *International Journal of Maritime Economics* 4, 126-148.
- Imai, A., E. Nishimura, K. Sasaki and S. Papadimitriou (2001). Solution comparisons of algorithms for the containerhip loading problem. *Proceedings of the International Conference on Shipping: Technology and Environment*, available on CD-ROM.
- Imai, A., K. Sasaki, E. Nishimura and S. Papadimitriou (2006). Multi-objective simultaneous stowage and load planning for a container ship with container rehandle in yard stacks. *European Journal of Operational Research* 171, 373-389
- Kang, J.-G. and Y.-D. Kim (2002). Stowage planning in maritime container transportation. *Journal of the Operations Research Society* 53(4), 415-426.
- Kim, K. H. (1994). Analysis of rehandles of transfer crane in a container yard. *APORS-Conference* 3, 357-365.
- Kim, K. H. (1997). Evaluation of the number of rehandles in container yards. *Computers & Industrial Engineering* 32(4), 701-711.
- Kim, K. H., J. S. Kang and K. R. Ryu (2004). A beam search algorithm for the load sequencing of outbound containers in port container terminals. *OR Spectrum* 26, 93-116.
- Kim, K. H. and D. Y. Kim (1994). Group storage methods at container port terminals. *The Materials Handling Eng. Division of 75<sup>th</sup> Anniversary Commemorative Volume ASME*, 15-20.
- Kim, K. H., Y. M. Park and K. R. Ryu (2000). Deriving decision rules to locate export containers in container yards. *European Journal of Operational Research* 124, 89-101.

- Lee, Y. and Y. J. Lee (2010). A heuristic for retrieving containers from a yard. *Computers & Operations Research* 37(6), 1139-1147.
- Mi, C., X. He, H. Liu, Y. Huang and W. Mi (2014). Research on a fast human-detection algorithm for unmanned surveillance area in bulk ports. *Mathematical Problems in Engineering* 2014.
- Mi, C., S. Yang, W. Mi and Y. Huang (2013). Ship identification algorithm based on 3d point cloud for automated ship loaders automated ship loaders. *Journal of Coastal Research* 73, 28-34. DOI: 10.2112/SI73-006.
- Mi, C., Z. Zhang., X. He, Y. Huang and W. Mi (2015). Two-stage classification approach for human detection in camera video in bulk ports. *Polish Maritime Research* 22 S1(86), 163-170.
- Ratcliffe, A. T. and P. Sen (1987). Computer aided stowage and lashings strategies for containerships. *Proceedings of Cargo Systems: Safe Ship-Safe Cargo Conference*, London, 113-117.
- Saginaw, D. J. and A.N. Parakis (1989). A decision support system for containership stowage planning. *Marine Technology* 26(1), 47-61.
- Sciomachen, A. and E. Tanfani (2007). A 3D-BPP approach for optimising stowage plans and terminal productivity. *European Journal of Operational Research* 183(3), 1433-1446.
- Shields, J. J. (1984). Container-ship stowage: a computer-aided preplanning system. *Mar. Technol.* 21(4), 370-383.
- Todd, D. S. and P. Sen (1997). A multiple criteria genetic algorithm for containership loading. *Proceedings of the Seventh International Conference on Genetic Algorithms*, 674-681.
- Wilson, I. D. and P. A. Roach (1999). Principles of combinatorial optimization applied to container-ship stowage planning. *Journal of Heuristics* 5, 403-418.
- Wilson, I. D. and P. A. Roach (2000). Container stowage planning: A methodology for generating computerised solutions. *Journal of the Operational Research Society* 51(11), 1248-1255.
- Wilson, I. D., P. A. Roach and J. A. Ware (2001). Container stowage pre-planning: Using search to generate solutions, a case study. *Knowledge-Based Systems* 14, 137-145.
- Winter, T. (1999). Online and real-time dispatching problems. Ph. D. Thesis, Technical University of Braunschweig, Germany.
- Yoke, M. H. L., X. Xiao, F. Liu, S. Y. Huang, W. J. Hsu and Z. Li (2009). An automated stowage planning system for large containerships. *Proceedings of the 4<sup>th</sup> Virtual International Conference on Intelligent Production Machines and Systems*.
- Zhang, W. Y., Y. Lin and Z. S. Ji (2005). Model and algorithm for container ship stowage planning based on bin-packing problem. *Journal of Marine Science and Application* 4(3), 30-36.
- Zhao, N. and Y. F. Shen. (2015). Vehicle-mounted task control system in container yard based on workflow engine. *Journal of Coastal Research* 73, 220-227.