

# Combinatorial Optimization and Strategy for Ship Stowage and Loading Schedule of Container Terminal

Wenbin HU<sup>1</sup>, Zhengbing HU<sup>2</sup>, Lei SHI<sup>1</sup>, Peng LUO<sup>1</sup> and Wei SONG<sup>1</sup>

<sup>1</sup>Wuhan University, Wuhan, China

<sup>2</sup>Huazhong Normal University, Wuhan, China

Email: hwb@whu.edu.cn

**Abstract**—In order to reduce the conflict of yard, vehicles and quay crane in the process of loading container, and upgrade safety of the ship stowage and transport economics, this paper took minimizing reshuffle times in yard and ship, minimizing the ship center of gravity, getting appropriate trim and minimizing the total moving distance of quay crane as optimization objectives to guarantee the balance of the ship and liquidity of vehicles as basic condition. It built ship stowage and loading scheduling model. The model propounded a ship loading scheduling strategy with heuristic greedy algorithm. The heuristic information in the heuristic greedy algorithm was the number and losses of the stacked container, bonus and fitness of the target container. The whole ship loading plan of container terminal was worked out by choosing the container having the least cost from yard during every schedule. The classification strategy and stacking strategy in bay were proposed in the ship stowage model. Based on these strategies, a scheduling model of ship stowage was established. A genetic algorithm which bases on group coding method and the stacking strategy in bay which bases on the greedy algorithm were proposed as the resolution of the ship stowage model. It can be concluded that combinatorial optimization algorithm and strategies were efficient and effective as the approach to ship stowage scheduling of container terminal from the analysis of the experiment's results.

**Index Terms**—Heuristic Information; Scheduling Strategy; Greedy Algorithm; ship Loading and stowage; GA

## I. INTRODUCTION

Nowadays, ship loading and stowage of container terminal is depended mainly on innate ship loading and stowage experiences to decide loading and stowage sequence. This method is too subjective to make effective use of the source of container logistics resources. For example, if minimizing the reshuffle times is the only factor to consider, and no attention paid to the total moving distance of quay crane, the quay crane will move frequently and the utilization of the facilities will be low. More ever, this situation may result in yard traffic problem because of vehicles continually fetching containers from container area. These phenomena would

decrease the efficiency of container terminal. Even worse, the depart time of the ship may be delayed which would damage the credit of container terminal. In order to increase the utilization of sources and decrease the conflict of yard, vehicles and quay crane, a proper ship loading plan is necessary to schedule. Ship stowage of container terminal is an important part of maritime transport. Ship stowage is a process before loading and unloading containers, which determines the stacking position of containers by following the basic stowage principles. The traditional stowage plan is carried out by human experience. But with the large-scale of container ships, stowage problem becomes more and more complex. The experience is no longer applicable. Shipping companies and various research institutions began to try to solve stowage problem by computer. Ship stowage is a complex combinatorial optimization problem with many factors. The stowage process needs to consider the safety and economy of transport. How to lower transport costs and improve operational efficiency on the basis of ensuring security is the primary problem for ship stowage.

Many researches were made to solve ship loading scheduling optimization problem. Literature [1] had put forward a heuristic optimization algorithm for external container reshuffle problem. It built an optimization model of the reshuffle with multiple hill-climbing algorithms, which reduced the reshuffle times effectively. Literature [2] had reduced the total reshuffle times by controlling the reshuffle times to one during each container schedule. It proposed two basic principles to minimize the cost of each container schedule. Literature [3] had combined minimal spanning tree algorithm and heuristic algorithm together, and fetched heavy container in priority during the ship loading process as basic principle to build crane fetching container optimal model with the goal of minimizing reshuffle times. The model got the optimal results based on minimal spanning tree generated by heuristic minimal spanning tree algorithm. The above research focus on minimizing reshuffle times, which considering the distribution of yard, allocation of vehicles and quay crane, balance of the ship and so on. The optimal result is hard to reach if only singular factor was taken into consideration in the process of ship

loading. Literature [8] had proposed a loading model of multi-type container and used COIN-OR to solve it. The container reshuffle rate reduced significantly. Literature [9] had proposed an optimal solution based on Pareto genetic algorithm in the premise of considering the safety and economy, which could increase the ship loading weight and reduced container reshuffle rate. Literature [10] had proposed a ship stowage strategy based on unit blocks, which could be a good solution to container sequence arrangement in case of the ship anchoring multiple ports. Literature [11] had proposed a ship stowage model for single bay and multiple ports, which aimed at minimizing the container reshuffle rate, but weight and other factors were not considered. The strategic decisions of current researches were focused on determining the specific stacking location for each container, but the number of containers and the scale of the problem increase rapidly with the large-scale of container ships. It is difficult to get a set of stowage program in a short time by the above strategies.

The above researches had considered a lot of different factors and optimized the ship loading process from diverse aspect. However, these researches hadn't considered the influence that the container loading sequence of quay crane in the yard reshuffle problem. Quay crane is a precious source in the ship loading process. Minimizing the moving distance of quay crane can increase the efficiency of ship loading process. And the proper container loading sequence can heavily decrease quay crane's moving distance. This paper takes minimizing yard reshuffle times and the total moving distance of quay crane as optimization objective, and takes allocation of vehicles and balance of ship as constraint condition. It builds ship loading scheduling model of container terminal. The model makes up a ship loading plan that had the least cost by using heuristic greedy algorithm. The heuristic information is not only the reshuffle times but also the dynamic changed top container type in yard and so on. The scheduling strategies can make the container logistic business more approximate to actual situation, which consider multivariate heuristic information in the ship loading. This paper proposes the classification strategy and stacking strategy in bay for ship stowage. On the basis of classification strategy, the concept of stacking unit is proposed. Based on summarizing the existing researches, this paper proposes the classification strategy and stacking strategy in bay. On the basis of classification strategy, the concept of stacking unit is proposed. The containers will be divided into different categories by classification strategy. It provides a great deal of flexibility for the following containers' loading plan and meets the requirements of real-time business of container

yard. The stacking strategy in bay takes into account the weight and discharge port of containers. It can guide the stacking of containers and improves the efficiency of the algorithm. On this basis, the mathematical model of ship stowage scheduling is established. The strategic decision is to determine the stacking location of each stacking unit. By this way, the scale of the problem reduces effectively. The goals are minimizing container reshuffle rate on board, minimizing the center of gravity of the ship, holding appropriate trim and ensuring that the heavy containers stacking in the middle of the ship. A genetic algorithm which bases on group coding method is proposed as the resolution of the model, ensuring the efficient and effective of ship stowage scheduling.

This paper is structured as follows: section 2 shows the scheduling model of ship loading of container terminal; section 3 presents greedy algorithm with heuristic information as ship loading schedule strategy; section 4 shows the schedule model of ship stowage of container terminal; section 5 presents the scheduling algorithm based on GA for ship stowage of container terminal; section 6 is experiment and results; section 7 is the conclusion and further works.

## II. MODEL OF SHIP LOADING SCHEDULE OF CONTAINER TERMINAL

### A. Ship Loading Schedule

#### 1) Premise of Ship Loading Schedule

This paper only considered ship loading scheduling of export container. All containers should gate into the terminals yard 3-5 days before ship loading. Considering the current yard space allocating situation and the constraint rule during ship loading process, for example, minimizing vehicles' driving distance and avoiding too many vehicles fetch containers from the same container area in yard to relieve the traffic of vehicles, Yard space allocating is a necessary. The yard distribution map can be formed after all containers gating into the terminals yard according yard space allocating plan. Another important premise of ship loading is ship stowage map. If the ship stowage map can satisfied the ship's stability and layout constraint, then it can be handed over to container terminal. Fig.1 shows the process of ship loading.

#### 2) Making the Scheduling Plan

Scheduling plan is ship loading sequence of container according to the yard space distribution map and ship stowage map, which makes up by heuristics greedy algorithm. Scheduling plan is made after the congregation period and before ship arriving. Some emergencies may

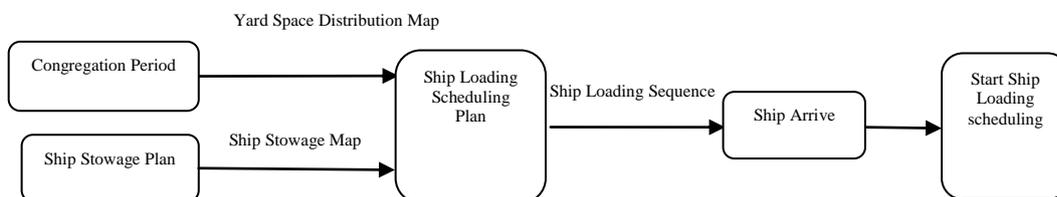


Figure1. Process of Ship Loading

result in the change of ship arriving time and the yard space distribution.

*B. Model of Scheduling Plan*

*1) Model Assumption and Definition*

Model Assumption: (1) Only having one ship in the container terminal at same time; (2) Only having one quay crane to service the ship at same time; (3) The container size assumes 40 feet.

Definition 1 (**Container Type**): The containers having the same category, size, destination and the weigh range are defined as a container type. The weigh range can influence the result of the algorithm.

Definition 2 (**Container Grid**): The volume of a container is defined as a container grid. Container grid is the unit for capacity measurement of ship and yard. For example, a ship with the size of 9\*4\*4 is 9 container grids long, 4 container grids width, 4 container grids high. The capacity of this ship is 144 container grids.

Definition 3 (**Quay Crane Moving Unit**): The width of ship bays is defined as quay crane moving unit. Quay crane moving unit equals to the width of a container grid.

Definition 4 (**Object Container**): The container in the yard that assumed to be fetched right away is defined as object container. If the object container has the least cost, it would fetch to the ship.

*2) Description of Scheduling Plan Model*

The decision content of ship loading plan includes: (1) Container loading sequence. The ship loading sequence of container would directly influence the yard reshuffle times and the traffic condition. (2) Containers' position on the ship. A container may have several positions to choose on the ship. The position of this container will influence quay crane's moving distance and stability of ship.

The decision goal of ship loading plan includes: (1) Minimizing the yard reshuffle times of the ship loading process; (2) Minimizing the total moving distance of quay crane.

The constraint conditions are:

(1) The total number of any container type in yard should greater more than the needed number of the container type on the ship.

(2) Vehicle sum of the same container section shouldn't exceed upper limit. In order to avoid the traffic jams of yard, vehicles should fetch container from different container section. The vehicle upper limit of in the same container section varies in different port. The upper limit assumes 5 in this paper.

(3) Guaranteeing the balance of ship. Weight of all containers assumes equal in this paper. In order to guarantee the balance of ship, the weight different between right and left of ship should within in a certain range. In this paper, the balance of ship is controlled by the ratio of the weight distance between right and left of ship. And the ratio varies with different ship shape and weight. This paper defined the upper limit of ratio as 0.8.

*3) Mathematical description of model*

Symbol Definition includes:

$C$  : Container section in yard;

$C_i$  : The  $i_{th}$  container section;

$C.number$  : Total number of container section in yard;

$C_i.CarNumber$  : The current sum of vehicles in the  $i_{th}$  container section;

$C_i.ContainerTypeNumber_j$  : The current sum of the  $i_{th}$  container type in the  $j_{th}$  container section;

$C_i.Ratio$  : Ratio of the needed container type and the total sum of this container type in the yard. The smaller the ratio, the bigger the reshuffle times in yard;

$S$  : Ship in the container terminal;

$S.Container$  : The sum of containers on  $S$  ;

$S.Box$  : The containers on  $S$  ;

$S.Box_i$  : The  $i_{th}$  container on  $S$  ;

$S.Box_i.Rolltime$  : The reshuffle times caused by  $S.Box_i$  ;

$S.Box_i.QuaysideMove$  : The moving distance of quay crane caused by  $S.Box_i$  when put it onto ship;

$S.LeftNumber$  : The total number of containers on the left side of ship.

$S.RightNumber$  : The total number of containers on the right side of ship.

$S.ContainerTypeNumber_j$  : The total number of the  $j_{th}$  container on ship at a moment.

Mathematical description of decision goal includes:

(1) Minimizing the yard reshuffle times during the whole ship loading process.

$$\min \left\{ \sum_{i=1}^{S.Container} S.Box_i.Rolltime \right\} \tag{1}$$

(2) Minimizing the total moving distance of quay crane.

$$\min \left\{ \sum_{i=1}^{S.Container} S.Box_i.QuaysideMove \right\} \tag{2}$$

Mathematical description of constraint conditions:

(1) The total number of any container type in yard should greater than the needed number of the corresponding container type on the ship.

$$S.ContainerTypeNumber_j \leq \sum_{i=1}^{C.Number} C_i.ContainerTypeNumber_j \tag{3}$$

(2) The sum of vehicles in the same container section shouldn't exceed upper limit.

$$C_i.CarNumber \leq 5 \tag{4}$$

(3) Guaranteeing the balance of ship.

$$\frac{|S.LeftNumber - S.RightNumber|}{(S.LeftNumber + S.RightNumber)} < 0.8 \tag{5}$$

III. GREEDY ALGORITHM WITH HEURISTIC FOR SHIP LOADING SCHEDULE STRATEGY

*A. Algorithm Description*

This paper used the greedy criteria in the process of using heuristic algorithm to solve ship loading scheduling model. The ultimate goal is to find the container having

the least cost from yard and fetch it to ship. Computing the cost of fetching a container from yard is the most important step in the problem solving process, which is computed based on heuristic information. Effective heuristic information can greatly reduce the reshuffle times and create favorable subsequent conditions, which can effectively reduce the subsequent reshuffle times and moving distance of quay crane. The following factors are the heuristic information:

(1) Reshuffle times caused by each fetching container from yard. All containers stacked on the object container need to reshuffle.

(2) Losses of stacked container. If there are containers stacked on the object container, the containers on the object container will be put on to the containers near the object container, which may cause reshuffle when the stacked container is fetched. Different stacked containers have different cost, which is the priority sequence of container inferred by the current ship loading situation. Sooner a container to be fetched, larger the cost be. Each container priority sequence calculates by the position of this container type on the ship. More in the bottom of ship and closer to quay crane, higher the container's priority sequence. Losses of stacked container are the sum of all stacked containers' cost caused by reshuffle.

(3) Bonuses after fetching container. The calculating method of bonuses is the same as losses calculating method. But their calculating objects are different. Bonuses' calculating objects are the containers under the object container. The losses calculating objects are the stacked container caused by reshuffle. If there are no containers under the object container, then the bonuses are zero.

(4) Integrated value of fetching container. Integrated value is an integrated function with reshuffle times, losses and bonuses. Integrated value = reshuffle times \* 5 + losses \* reshuffle times - bonuses.

(5) Fitness of fetching container. Fitness is the most nearest distance of quay crane during each fetching container to ship. If integrated value of several container types is the same, it chooses the type of container with the smallest fitness first considering the condition of keeping the stability of ship, which can decrease greatly the total moving distance of quay crane.

Literature [4] pointed out that quay crane has its work section. Quay crane should only work in its work section, and quay cranes should keep a certain distance with each other. This paper assumes that only one quay crane serves the ship, so the quay crane can move to any bay on ship. The fetched container should only put on the current bay or two bays around the current bay on the ship. This strategy can guarantee that quay crane just need move one bay for each operation, so that the total moving distance of quay crane can be decreased. This paper uses greedy algorithm to determine which container grid would be placed which kind of container type, which has the least cost and satisfies the constraints from the top container grid of current bay and the two bays around the current bay. When calculating the cost of one container type, it assumes that the container type is type *A*. There

are several *A* containers in yard. According to greedy criteria, the container of type *A* having the least yard cost would be selected.

### B. Algorithm Pseudo Code

Input: ship storage map and yard space distribution map.

Output: container loading sequence.

1. Initializing the position of quay crane, total number of vehicles, the value of *least cost*, *bonus*, *loss*, *fitness*, Integrated value *com* and so on;

2. While Ship is not filled;

3. For Container grid in the current bay or two bays around on ship;

4. If This container grid is not filled and can guarantee ship's stability after ship filled. It marks the container's type as *A* according to the ship storage map;

5. For Every container section *CS<sub>i</sub>* in yard;

6. If Container section *CS<sub>i</sub>* has the container type *A* and the total number of vehicles in *CS<sub>i</sub>* doesn't exceed the upper limit;

7. For Every container *C<sub>j</sub>* in the container section *CS<sub>i</sub>*;

8. If Container *C<sub>j</sub>*'s type is *A*, calculating the total number of containers above the container *C<sub>j</sub>*, *least cost*, *bonus*, *loss*, *fitness*, *com* and so on;

9. For Every container *PB<sub>k</sub>* above the container *C<sub>j</sub>*

10. Calculating the value of *loss<sub>k</sub>*, losses of putting container *PB<sub>k</sub>* onto the around container. There may be several empty container grids in yard for container *PB<sub>k</sub>*, the container grid having the least *loss<sub>k</sub>* would be chose to put the container. *loss* = *loss* + *loss<sub>k</sub>*;

11. REPEAT

12. Calculating *bonus* of container *C<sub>j</sub>*, then calculating *com<sub>i</sub>* and *fitness<sub>i</sub>* of container *C<sub>j</sub>*. If *com<sub>i</sub>* is smaller than *com*, or *com<sub>i</sub>* equal to *com* and *fitness* smaller than *fitness<sub>i</sub>*, it assigns the value of *com* to *com<sub>i</sub>* and assigns the value of *fitness* to *fitness<sub>i</sub>*;

13. REPEAT

14. REPEAT

15. REPEAT

16. If An empty container grid which can place a container on ship is found, the container grid is placed on ship and the corresponding container in yard is added to the ship loading sequence. It updates the ship stowage map and yard space distribution map, if an empty container grid which can place a container on ship can't be found, then move quay crane a unit left or right;

17. REPEAT

## IV. MODEL OF SHIP STOWAGE SCHEDULE OF CONTAINER TERMINAL

### A. Abstraction and Analysis of Ship Structure

This paper abstracts the ship's container stacking space. The abstracted stacking space is a rectangular space and

each container will be stacked in the appropriate place in the rectangular space. Considering the stacking space's shapes of different ships are different, this paper abstracts the stacking space as a rectangular space in order to improve the versatility of the scheduling model constructed. The rectangular space can be extended into any other irregular space by hierarchically in the actual operation.

Fig.2 shows the ship's bay graph. The left figure is the actual ship' bay graph and the right figure is the abstracted ship's bay graph. For different ships, the stacking spaces on the deck are the same and they are rectangular spaces. However, stacking space under the deck is involved in specific structure of the ship. The model in this paper adopts a rectangular space shown on the right. The universal of this space mainly reflects in three aspects. (1) We can add some constraints, such as prohibiting some slots for stacking, thus the stacking space in the right figure can be transformed into the stacking space in the left figure. (2) It doesn't need to transform the stacking space in the right figure. Keep the relative stacking layers when stack the containers. (3) The stacking space of the whole ship can be divided into several subspaces for stacking. The subspaces are all rectangular spaces. We can take the model and algorithm in this paper to solve the subspaces.

*B. Classification Strategy and Stacking Unit*

**Definition 5(container group):** A container group [5] means containers whose size, type, loading ship, unloading port and weight level are the same.

In most researches of the container ship stowage scheduling, the strategic decision is to specify each container's specific stacking position on board. It's effective when the number of containers is smaller. With large-scale of the ships and increasing of the number of containers involved in the stowage plan, more shortcomings come out. Overall, this approach has two following prominent weaknesses: (1) the optimization space is too large; (2) each container's stacking position is specified when the stowage plan is completed, which strictly limits the loading sequence of containers and is not conducive to meet the requirements of real-time business. The idea of the container classification is proposed in this paper. Based on the classification of containers, the containers will be divided into stacking

unit.

**Definition 6 (classification strategy):** Classification strategy is built based on container group. (1) If the weights of containers in a container group are not very different, the containers in the container group will be designated as a category of container. (2) If the weights of containers in a container group are very different, the containers will be clustered based on the weight to form several categories of container.

**Definition 7 (stacking unit):** Stacking unit is a set spliced by some containers which belong to a container category and occupying a layer of slots in a bay.

The strategic decision of this paper is to determine the location of stacking unit. This approach can greatly reduce the solution space. At the same time, since only the stacking position of stacking unit is determined (i.e. only the stacking positions of the container categories are determined and the specific location of each container is not determined), it can bring a lot of flexibility for the following ship loading plan. The containers in the same stacking unit belong to a same category of containers, so the weight is not very different. Since the stacking unit occupies all the slots in a layer of a bay, the balance between the left and right of the ship does not need special consideration. In the actual loading process, using some simple strategies can ensure the balance between the left and right of the ship.

Formation of stacking unit: Suppose the number of containers in a container category is  $n$  and the number of containers which can be stacked into a layer of a bay on board is  $N$ . Suppose the quotient of  $n/N$  is  $x$  and the remainder is  $y$ . When  $y < N/4$ , the containers of this category will be divided into  $x$  stacking units. The  $y$  remaining containers will not be included in any stacking unit. They will be suitably stacked in the end of the process. When  $y > N/4$ , the containers of this category will be divided into  $x + 1$  stacking units. One of these units has only  $y$  containers.

According to the strategy of stacking unit formation, the number of containers in some stacking units will be less than  $N$ , which means that the stacking unit can't take all the slots in a layer of a bay. But the algorithm still thinks that these stacking units occupy a layer of slots in a bay. In the actual loading process, containers which are

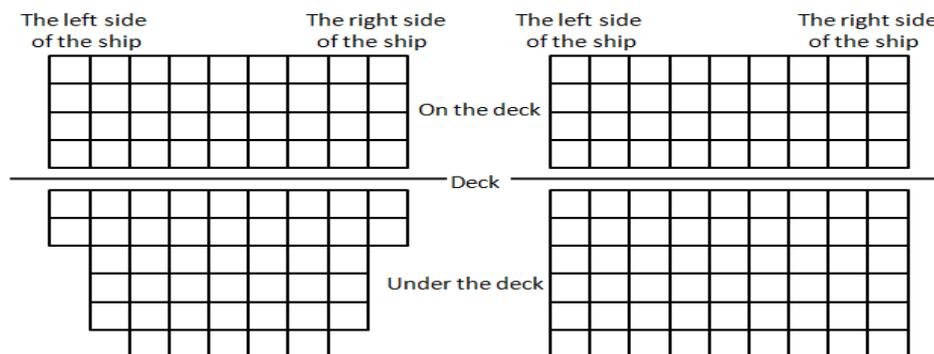


Figure2. The ship's bay graph

not included in any stacking unit can be assigned to these slots. Containers in stacking unit which stacks in the higher layer can be also assigned to these slots.

C. Stacking Strategy in Bay

There are two main factors which have an effect on the container stacking sequence in a bay. One is the discharge port of the stacking unit and the other is the weight of the stacking unit. The general principle of stowage is to ensure the heavy stacking unit stacks below the light stacking unit to get a lower centre of gravity. Also, it tries to ensure the stacking unit with a farther discharge port stacks below the stacking unit with a nearer discharge port, which can turns down the container reshuffle rate. But the two principles are difficult to be met at the same time in practice. If we only consider the weight and strictly obey the rule that heavier containers stacks below the lighter containers, it will inevitably lead to the whole ship's high rate of container reshuffle. If we only consider the discharge port and strictly obey the rule that containers with a farther discharge port stacks below containers with a nearer discharge port, it will inevitably lead to a higher center of gravity and affects the stability of the ship. A greedy strategy is proposed to guide the stacking of stacking units in a bay in this paper. This strategy takes into account the discharge port and the weight of stacking units. Define the discharge port of a stacking unit as  $P$ , and weight as  $W$ .  $P = j$  means the  $j^{th}$  port the ship will arrive after leaving the current port. Define a variable  $wMULp = w^\alpha * p^\beta$ .  $\alpha$  is the weight of  $W$  and  $\beta$  is the weight of  $P$ .

Definition 8 (stacking strategy in bay): The stacking units in a bay will be stacked by the value of  $wMULp$ . Stacking unit with a larger value of  $wMULp$  will be stacked in the lower layer, and stacking unit with a smaller value of  $wMULp$  will be stacked in the higher layer.

Fig.3 and Fig.4 show the values of container reshuffle rate and center of gravity under different values of  $\alpha$  and  $\beta$ .

Center of gravity is the relative center of gravity. The calculation is: relative center of gravity = (Absolute center of gravity – the min value of center of gravity)/the difference between the max and min value of center of gravity. Container reshuffle rate and the center of gravity are both related to  $\alpha / \beta$ . When  $\alpha / \beta$  increases, the container reshuffle rate will increase and the relative center of gravity will decrease. When  $\alpha / \beta$  decrease, the container reshuffle rate will decrease and the relative center of gravity will increase. Stacking strategy in bay proposed in this paper can guide the stacking of stacking units efficiently, which will improve the efficiency of the algorithm. In actual stowage process,  $\alpha / \beta$  can be adjusted according to the practical requirements.

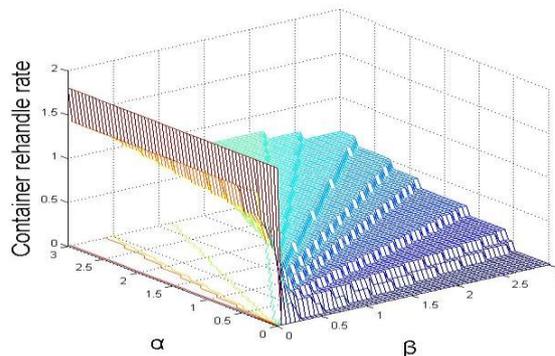


Figure 3. The values of container reshuffle rate under different  $\alpha$  and  $\beta$

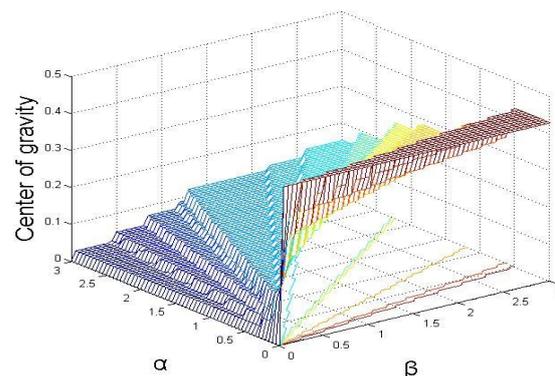


Figure 4. The values of center of gravity under different  $\alpha$  and  $\beta$

D. The Description of Stowage Scheduling Model

Considering the stacking of refrigerated containers and special containers has other constraints, the model in the  $\alpha$  and  $\beta$  paper is only for general export containers. The strategic decisions of this paper are that which bay each stacking unit is assigned and the stacking location of each stacking unit in bay. The goals are as follows:

- (1) Minimize the rate of container reshuffle on board. The cost of container reshuffle on board is high. Decreasing the container reshuffle can get good economic benefits.
- (2) Minimize the centre of gravity. A lower centre of gravity allows the ship to maintain a good stability and ensure the safety of navigation.
- (3) Meet the rule that heavy containers must stack in the middle of the ship. It can prevent the arch of the ship's middle part.
- (4) Maintain proper head and tail trim of the ship. The proper trim can make the navigation of the ship has good performance.

Different ships have different requirements for trim. For the convenience of presentation, this paper agrees that 0 is the best value of trim. In practice, we can adjust the evaluation function of the trim to get the perfect trim for the specific ship according to the specific requirements of the ship.

E. Mathematical Description of the Model

1) The definition of mathematical symbols

*BayNum*: The number of bays on board;

*TierNum*: The number of containers which can be stacked in a layer of a bay;

*ContainerNum*: The number of containers to be shipped;

*TierNumBay<sub>i</sub>*: The maximum number of layers in the *i<sup>th</sup>* bay. Considering the stacking space is a rectangular space, so *TierNumBay<sub>i</sub>* = *TierNumBay<sub>j</sub>* (*i* ≠ *j*);

*BayW<sub>i</sub>*: The total weight of containers stacked in the *i<sup>th</sup>* bay;

*w<sub>ij</sub>*: The total weight of containers stacked in the *j<sup>th</sup>* layer of the *i<sup>th</sup>* bay.

*h<sub>j</sub>*: The distance between the stacking unit in the *j<sup>th</sup>* layer and the stacking flat.

*l<sub>i</sub>*: The distance between the *i<sup>th</sup>* bay and the ship's center. The value of distance between the center and bay which is close to the ship's head is positive. The value of distance between the center and bay which is close to the ship's tail is negative.

*port<sub>ij</sub>*: The discharge port of stacking unit in the *j<sup>th</sup>* layer of the *i<sup>th</sup>* bay. Discharge port is defined in digital form. Suppose a ship has 5 discharge ports, then *port<sub>ij</sub>* ∈ {1, 2, 3, 4, 5}. *port<sub>ij</sub>* = 1 Means the nearest port from the current port, and *port<sub>ij</sub>* = 5 means the farthest port from the current port.

*TierUsedInBay<sub>i</sub>*: The number of used layers in the *i<sup>th</sup>* bay.

*Re HandleNumInBay<sub>i</sub>*: Reshuffle number in the *i<sup>th</sup>* bay. It can be calculated by equation (6).

$$Re\ HandleNumInBay_i = \sum_{j=1}^{TierUsedInBay_i-1} \sum_{k=j+1}^{TierUsedInBay_i} (a_{ijk} * TierNum) \tag{6}$$

$$a_{ijk} = \begin{cases} 0, & port_{ij} \geq port_{ik} \\ 1, & port_{ij} < port_{ik} \end{cases} \tag{7}$$

*WeightCenter<sub>i</sub>*: The Center of gravity of all the stacking units in the *i<sup>th</sup>* bay. It can be calculated by equation (8).

$$WeightCenter_i = \frac{\sum_{j=1}^{TierUsedInBay_i} w_{ij} * h_j}{\sum_{j=1}^{TierUsedInBay_i} w_{ij}} \tag{8}$$

*Center*: The evaluation value of heavy containers stacked in the middle. It can be calculated by equation (9).

$$Center = \sum_{i=1}^{BayNum} c_i \tag{9}$$

*c<sub>i</sub>* Can be calculated by

$$c_i = \begin{cases} 1, & \text{if } (BayW_i \leq BayW_{centerp}) \\ 0, & \text{if } (BayW_i > BayW_{centerp}) \end{cases}, \text{ centerp} \in CENTER,$$

*CENTER* is the set of bays in the center of the ship.

*Trim*: The inclusive trim of a ship. It can be calculated by equation (10).

$$trim = \frac{|\sum_{i=1}^{BayNum} l_i * BayW_i|}{\sum_{i=1}^{BayNum} |l_i| * BayW_i} \tag{10}$$

2) Goals of mathematical description of the model

(1) Minimize the container reshuffle rate on board.

$$\min\left(\frac{\sum_{i=1}^{BayNum} Re\ HandleNumInBay_i}{ContainerNum}\right) \tag{11}$$

(2) Minimize the average centre of gravity.

$$\min\left(\frac{\sum_{i=1}^{BayNum} WeightCenter_i}{BayNum}\right) \tag{12}$$

(3) To ensure that the heavy containers stacked in the middle of the ship.

$$\max(center) \tag{13}$$

(4) To maintain proper head and tail trim.

$$\min(trim) \tag{14}$$

3) Constraints of mathematical description of the model

Stacking layers should not exceed the maximum stacking layers of each bay.

$$TierUsedInBay_i \leq TierNumBay_i \tag{15}$$

V. GA-BASED CONTAINER TERMINAL SCHEDULING ALGORITHM FOR SHIP STOWAGE

A. Encoding of Genetic Algorithm

The encoding of genetic algorithm will directly affect the performance of the algorithm. Most literatures solve stowage problem with genetic algorithm based on container-based encoding. The length of the encoding string is the number of containers. This encoding is applicable when the number of containers is less. But the number of containers is large in the ship stowage at present. This encoding is clearly no longer applicable. Too long code length will seriously affect the efficiency of the algorithm. Literature [12] used genetic algorithm with group coding to solve bin-packing problem, overcoming the shortcomings of traditional coding methods. This paper also adopts this encoding method. At the same time, the stacking strategy in bay will be used to guide the stacking of each stacking unit.

The genetic information expresses that which units are assigned to which bay. The stacking strategy in bay will guide the stacking of stacking units in a bay.

Chromosome’s length (i.e. number of genes) is the number of bays on board. A gene represents a bay. A chromosome which represents 10 bays is shown in Fig. 5. Every gene stores the information of stacking units which are assigned to the corresponding bay. The numbers in each gene are the serial number of stacking units.

**B. Description of Genetic Operators**

**1) CROSSOVER OPERATOR**

In this paper, the chromosomal level and gene level crossover operator are used. Chromosomal level crossover operator’s iterative process is shown in Fig.6. The iterative steps are as follows:

(1) Select two chromosomes from the population, then randomly select two crossover points. The genes between the two crossover points are to be exchanged. They are named Genes\_1 and Genes\_2.

(2) Construct several sets Sf, Ss, F and S. Sf is the set which consists of stacking units in Genes\_1.

Ss are the set which consists of stacking units in Genes\_2. F and S are defined by the following equations.  $F = \{a | a \in Sf \wedge a \notin Ss\}$ ,  $S = \{a | a \in Ss \wedge a \notin Sf\}$ .

(3) Exchange Genes\_1 and Genes\_2.

(4) Start the validity check. For every stacking unit a in Genes\_1, if  $a \in F$ , then select stacking unit b from S to replace a. The stacking unit selected from S must be different each time. If no more stacking units can be selected from S, then remove the stacking unit a. Check the whole chromosome which includes Genes\_1 and assign the stacking units which don’t appear to some

genes randomly. If  $c \in S$  in every stacking unit c in Genes\_2, then select stacking unit d from F to replace c. The stacking unit selected from F must be different each time. If no more stacking units can be selected from F, then remove the

stacking unit c. Check the whole chromosome which includes Genes\_2 and assign the stacking units which don’t appear to some genes randomly.

Gene level crossover operator: Randomly select two genes in a chromosome and get the information of stacking units in them. Then assign these stacking units into the two genes randomly.

**2) MUTATION OPERATOR**

Randomly selecting a gene and its neighbor in a chromosome, then exchange these two genes’ information (i.e. exchange the information of stacking unit in the two genes).

**3) SELECTION OPERATOR**

Roulette wheel selection operator is used.

**C. Algorithm Steps**

(1) Initialize population. The stacking units will be assigned into bays randomly. Calculate the value of  $wMULp$  for each stacking unit. Guide the stacking of stacking unit by stacking strategy in bay proposed in this paper. Turn to (2).

(2) Evaluate each individual with the business rules in actual stowage, and figure out the current best individual. Turn to (3).

1	11	2	15	3	4	6	5	7	8
10	12	13	16	17	18	19	21	25	9
31	32	14	33	36	38	20	22	26	27
46	43	45	34	37	39	30	23	29	28
47	44	48	35	49	50	40	24	41	42

Figure5. Chromosome diagram

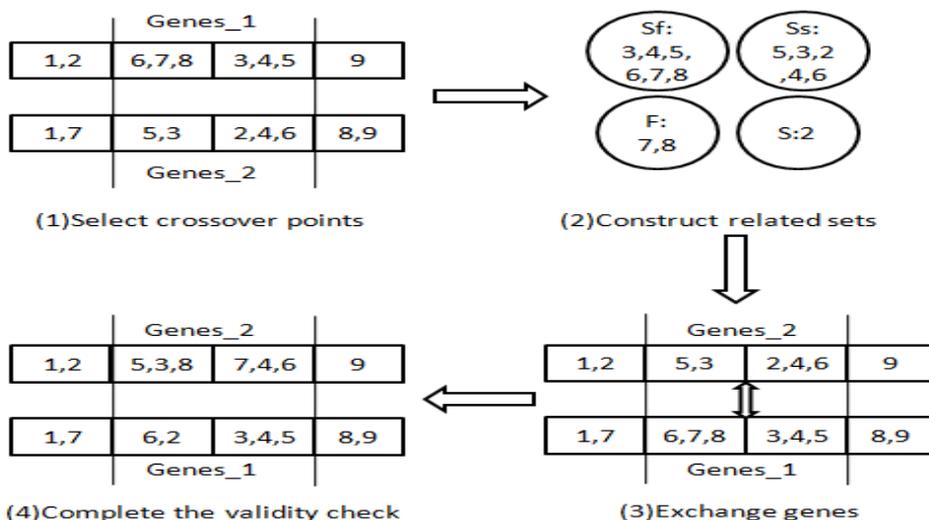


Figure6. Schematic diagram of chromosomal level crossover operator

(3) Judge whether the algorithm meets the condition for ending or not. Number of iterations is the ending conditions of the algorithm. If the algorithm ends, then turn to (8), otherwise, turn to (4).

(4) Execute selection operator. Roulette wheel selection operator is used. Turn to (5).

(5) Execute crossover operator. First, execute chromosomal level crossover operator. Start the validity check. Second, execute gene level crossover operator. Turn to (6).

(6) Execute mutation operator. Turn to (7).

(7) Calculate the value of  $wMULp$  for each stacking unit. Guide the stacking of stacking unit by stacking strategy in bay. Evaluate each individual with the business rules in actual stowage, and figure out the current best individual. Execute keep-alive strategy for the elite. Turn to (3).

(8) The iteration ends. Print the optimum solution at present and explain the solution according to the business rule. Turn to (9).

(9) The algorithm ends.

VI. EXAMPLES AND RESULTS ANALYSIS

A. Ship Loading Experiments

1) EXPERIMENTAL HYPOTHESIS

This experiment assumes that the size of the ship is  $9*4*4*$ , with 9 unit of length, 9 unit of width, 4 unite of height. The size of yard is  $4*3*5*5*4$ . The yard has  $4*3$  container area. Each container area has 5 unit of length, 5 unit of width, 4 unit of height. It supposes there are 10 types of container on ship. The container types number respectively from 0 to 9. The amount and position of containers on ship are randomly generated. The amount of container type in yard can be freely set. It may greater than or equal to 10, as long as there are enough containers in yard to meet the ship's requirement. The amount of container type in yard will affect the total reshuffle times. Fig.7 shows the reshuffle times and moving distance of quay crane calculated out by heuristic greedy algorithm, when the amount of container type in yard is respectively taken 30, 40, 50, 60 and 70.

It can be seen from the experiment result to five different container types in yard that the reshuffle times in yard is linear increased and the moving distance of quay crane is decreased gradually with the increasing of the amount of container type in yard. Although small number of container types in yard can decrease the reshuffle times. But it would increase the moving distance of quay crane.

The types of containers are not completely randomly generated when initializing the yard. Literature [5] [6] respectively used genetic algorithm and reserving-based strategy to guide the container area and bay allocation before containers were stored in yard. When containers reached the yard, Literature [7] used mixed-row-based space allocation strategy found the set of plan bays according to the containers' ship. If an empty container grid meeting the optimization goal was found in the set of

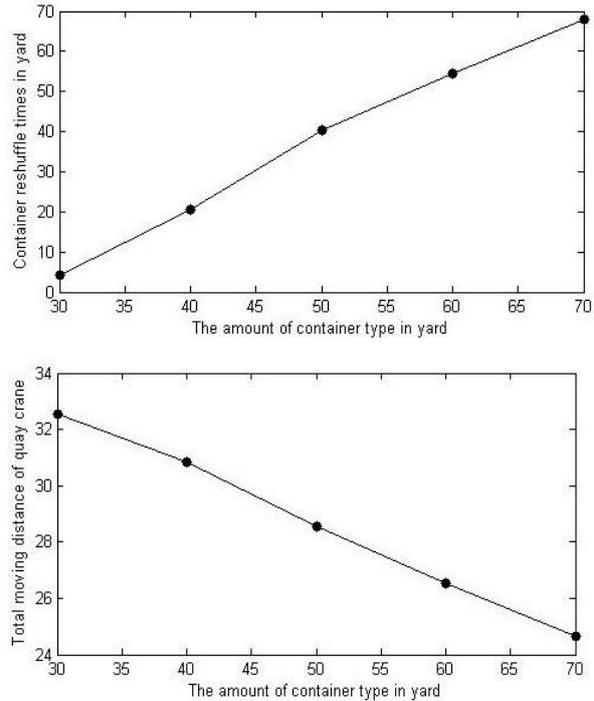


Figure7. Heuristic Greedy Algorithm's optimal result with different container type

plan bays, then put the corresponding container in that container grid. It can be seen that the containers belonging to the same ship are relatively concentrated in yard and most of the containers are located in the planed bays. Therefore, this study polymerized the containers of the same ship when initializing the yard so that the containers of the same ship are relatively concentrated in yard. The not completely random initialization method is randomly generating a container type first at the beginning of initializing yard. If this container type belongs to ship, the next container type would be generating must belongs to ship. After that, generating a container type randomly and then do as the above method until the end of initializing yard. The initializing result should meet to (3) any time in the initializing process. Fig.8 shows the reshuffle times and moving distance of quay crane with different container type amount in yard when the yard is initialized completely random and not completely random.

Fig.8 shows that the yard reshuffle times calculating out by heuristic algorithm turning a big difference under the condition of yard completely randomly initializing and not completely randomly initializing. The reshuffle times of yard completely randomly initializing is 46.98% to 74.23% greater than the reshuffle times of yard not completely randomly initializing. But the moving distance of quay crane are almost the same under the two conditions. The comparison experiment in this paper would use the realistic yard not completely randomly initializing method to initialize the yard.

2) EXPERIMENT COMPARISON

(1) The comparison between heuristic greedy algorithm and randomly fetching container method

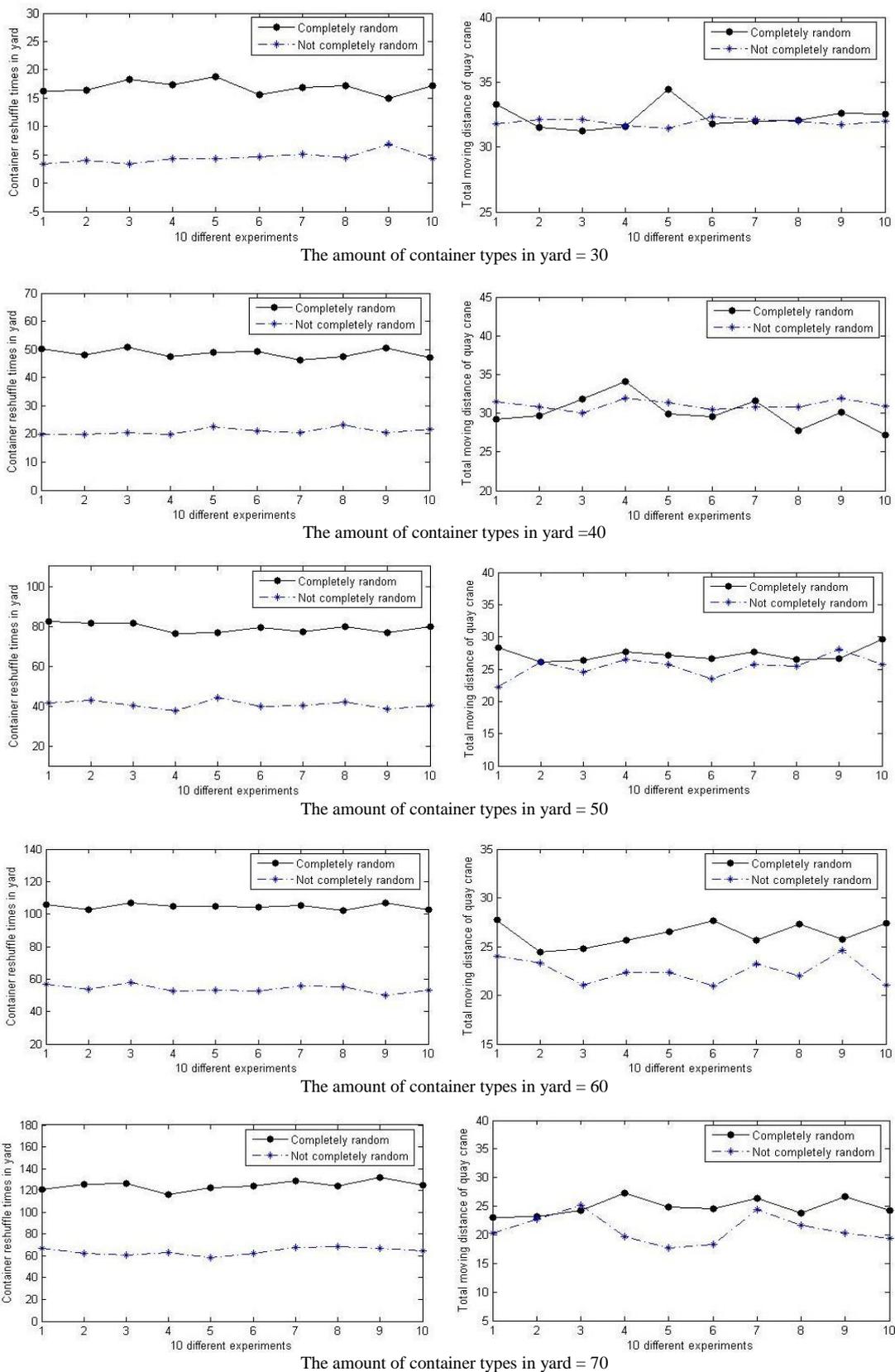


Figure8. The comparison of vard completely initializing and not completely initializing

The ship loading process of completely randomly fetching container method is to load ship from bottom to up, from left to right. It fetches the container to meet the type requirement from any position of yard. The reshuffles don't consider any heuristic information in

yard and the containers above the object container can move to any empty container grid around. Table I shows the reshuffle times and moving distance of quay crane of the heuristic greedy algorithm and completely randomly getting container method in 10 different experiments.

TABLE I .  
THE COMPARISON BETWEEN THE HEURISTIC GREEDY ALGORITHM AND COMPLETELY RANDOMLY FETCHING CONTAINER METHOD

Observed parameters	Heuristic greedy algorithm		Completely random getting container method	
	Reshuffle times	Moving distance of quay crane	Reshuffle times	Moving distance of quay crane
	53.34	26.06	127	56
	53.76	24.76	150	56
	55.86	26.58	116	56
	54.75	24.8	174	56
	55.44	25.94	159	56
	50.12	27.08	149	56
	56.52	27.42	132	56
	53.6	24.69	158	56
	53.93	25.00	144	56
	50.88	26.55	134	56
Average	53.82	25.88	144.3	56

Table I show that the reshuffle times of heuristic greedy algorithm is decreased by 62.70% than the completely randomly fetching container method. At the same time, the moving distance of heuristic greedy algorithm is decreased by 53.79% than the completely randomly fetching container method. It can be seen that the heuristic greedy algorithm's performance is more superior to the custom completely random getting container method.

(2) The comparison between heuristic greedy algorithm and greedy algorithm without heuristic information

The greedy algorithm without heuristic information just only considers the reshuffle times, but not considers the bonuses, losses and fitness, when calculating the cost of getting container in yard. The heuristic greedy algorithm considers all the heuristic information when calculating the cost. It can be seen the guide significance of the heuristic information in heuristic greedy algorithm by the comparison of two algorithms. Fig. 9 shows the reshuffle times and moving distance of quay crane of heuristic greedy algorithm and greedy algorithm without heuristic information under different container types in yard.

Fig.9 shows that the reshuffle times of heuristic greedy algorithm is decreased by 8.05% to 50.24% than the greedy algorithm without heuristic information under the same condition. In the same time, the moving distance of quay crane is decreased by 21.5% to 38.29%. It can be seen that the heuristic information can help to reduce the reshuffle times and moving distance of quay crane.

## B. Ship Stowage Experiments

### 1) EXPERIMENTAL HYPOTHESIS

The experiments will be carried out in situations of 50 bays, 60 bays, 70 bays, 80 bays, 90 bays and 100 bays on board. Suppose all the containers are 40-foot containers (In practice, two 20-foot containers make up a 40-foot containers.). The number of layers in each bay on board is 10. The ship anchored 5 port (Does not include the starting port). Consider the situation that assign  $10 * n$  stacking units into  $n$  (the value of  $n$  can be 50, 60, 70, 80, 90 and 100) bays. The weight of stacking unit varies

from 5 to 30. Weight given here is the relative weight of stacking units, rather than the actual weight.

The numerical experiments will be carried out in two parts. The first part validates the model and algorithm proposed in this paper. The second part shows different optimization results under different values of two parameters (i.e.  $\alpha, \beta$ ) of the stacking strategy.

### 2) THE VALIDATION OF THE MODEL AND ALGORITHM

Set the crossover rate to be 0.9, mutation rate to be 0.1 and iterations to be 1000. In case of random stacking and using the model and algorithm proposed in this paper, we can get the results shown in Table II. The parameters ( $\alpha, \beta$ ) are set to be 3 and 10 respectively.

On the analysis of the data in Table II, we conclude that the relevant indicators have obtained better results after optimized by using the algorithm in this paper. Fig.6 shows the weight distribution on 50-bays ship in the two above cases. We can get the following conclusions from Fig.6. In case of using the optimization algorithm, the weight of containers stacking in middle bays (the 22-29 bay) is significantly greater than other bays, meeting the requirements of heavy containers stacking in the middle. In case of completely random stacking, the weight of containers stacking in middle bays is not very different from other bays. It didn't meet the requirements of heavy containers stacking in the middle and the arch of the middle part of the ship was easy to happen. The results of 60-100 bays are similar to the results of 50 bays, so we don't list them here.

On the analysis of the data in Table II, we conclude that the relevant indicators have obtained better results after optimized by using the algorithm in this paper. Fig.6 shows the weight distribution on 50-bays ship in the two above cases. We can get the following conclusions from Fig.6. In case of using the optimization algorithm, the weight of containers stacking in middle bays (the 22-29 bay) is significantly greater than other bays, meeting the requirements of heavy containers stacking in the middle. In case of completely random stacking, the weight of containers stacking in middle bays is not very different from other bays. It didn't meet the requirements of heavy containers stacking in the middle and the arch of the middle part of the ship was

easy to happen. The results of 60-100 bays are similar to

the results of 50 bays, so we don't list them here.

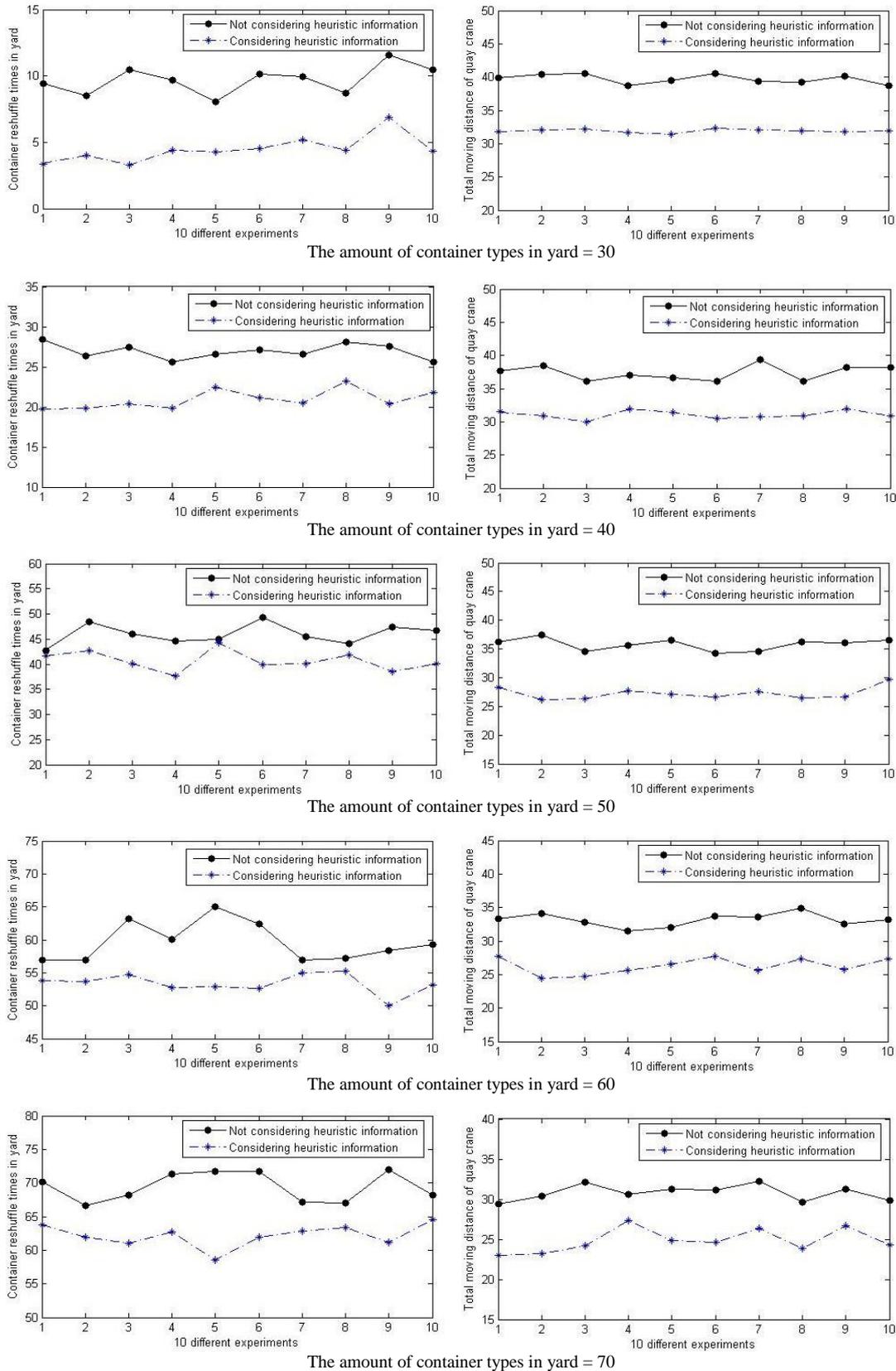


Figure9. The comparison between heuristic greedy algorithm and greedy algorithm without heuristic information

TABLE II  
THE COMPARISON BETWEEN THE RESULT OF RANDOM STACKING AND THE RESULT OF OPTIMIZATION

The number of bay	Completely random stacking			Using the algorithm and strategy		
	Container remanded rate	Center of gravity	Trim	Container reshuffle rate	Center of gravity	Trim
50	181.6%	0.4967	0.015045	0%	0.1953	0
60	173%	0.5303	0.038356	0%	0.1551	0
70	182%	0.5065	0.021358	0%	0.2049	0.000005
80	176.25%	0.4967	0.042131	0%	0.2030	0.000060
90	180.89%	0.5011	0.045426	0%	0.1760	0.000027
100	179.3%	0.4856	0.048069	0%	0.2035	0.000012

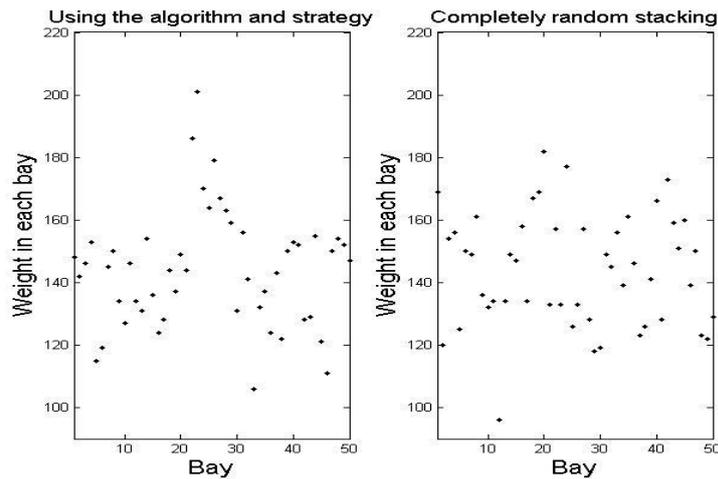


Figure10. The comparison of weight distribution in 50-bays ship

3) OPTIMIZATION RESULTS UNDER DIFFERENT VALUES OF  $\alpha$  AND  $\beta$

On the analysis of stacking strategy in bay, we can conclude that the optimized value is independent of the absolute value of  $\alpha$  and  $\beta$  but related to their relative value. We will discuss the optimization results in case of  $\alpha / \beta$  ranges from 0-1 and the optimization results in case of  $\beta / \alpha$  ranges from 0-1.

When  $\alpha / \beta$  ranges from 0-1, set the crossover rate to be 0.9, mutation rate to be 0.1 and iterations to be 1000. We get the experimental results shown in Fig. 11 and Fig. 12.

On the analysis of Fig.11 and Fig.12, we can conclude that the container reshuffle rate has remained at a very low level and the largest rate is less than 6%. Relative centre of gravity remains at 0.1-0.36 range. Container reshuffle rate and the centre of gravity both obtain good optimization values when  $\alpha / \beta$  in the 0.1-0.4 range.

When  $\beta / \alpha$  ranges from 0-1, set the crossover rate to be 0.9, mutation rate to be 0.1 and iterations to be 1000. We get the experimental results shown in Fig13 and Fig14.

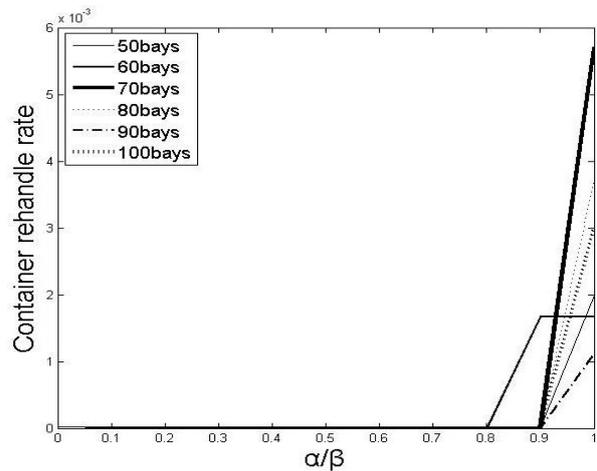


Figure11. Container reshuffle rate under different values of  $\alpha / \beta$

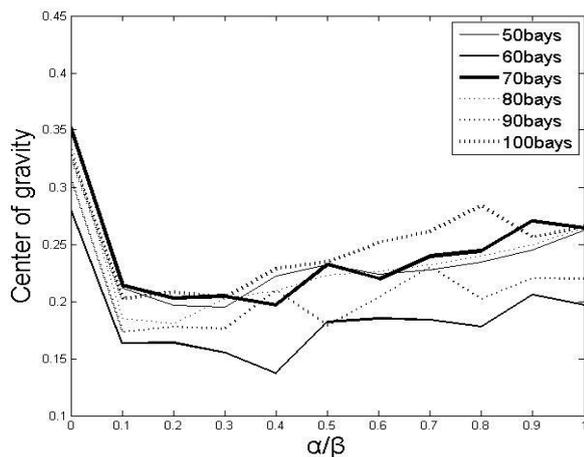


Figure12. Relative centre of gravity under different values of  $\alpha / \beta$

On the analysis of Fig13 and Fig14, we can conclude that with the increase of  $\beta / \alpha$ , container reshuffle rate decreases and the centre of gravity increases. Container reshuffle rate and the centre of gravity both obtain good optimization values when  $\beta / \alpha$  in the 0.7-1 range.

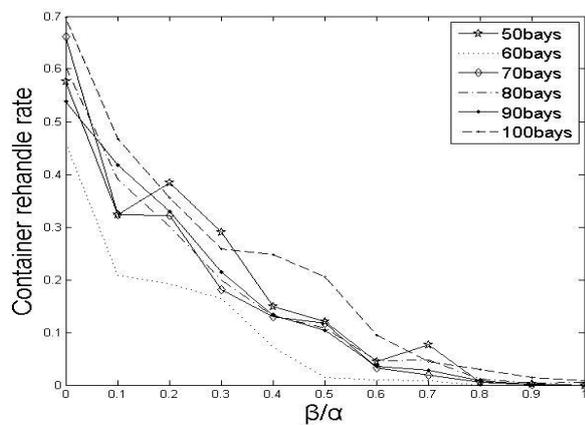


Figure13. Container reshuffle rate under different values of  $\beta / \alpha$

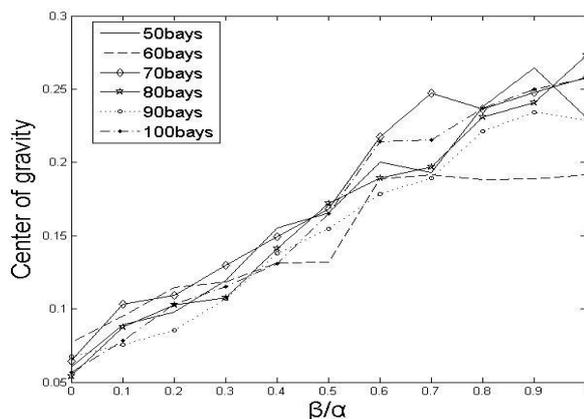


Figure14. Relative centre of gravity under different values of  $\beta / \alpha$

## VII. CONCLUSION AND OUTLOOK

According to the ship storage map and yard space distribution map, This paper took minimizing reshuffle times in yard and the total moving distance of quay crane as optimization object to guarantee the balance of the ship and liquidity of vehicles. It built a ship loading scheduling model of container terminal. The model can make the whole container ship loading plan by using the heuristic greedy algorithm to choose the container having the least cost in yard during every loading. Many experiments have verified the utility of the heuristic greedy algorithm. Based on this paper, subsequent research can take the allocation of bridge crane, routes choice of vehicles and other factors into account in the cost evaluation function. In the same time, subsequent research can consider the situation of having multiple quay cranes on a ship and combine the scheduling of quay crane with the heuristic greedy algorithm in this paper. Then the ship loading sequence can be more practical. Ship stowage scheduling is a typical combinatorial optimization problem. With the large-scale of container ships, the number of containers increases, thus increasing the difficulty of solving such problems. The concept of stacking unit is proposed in this paper. The strategic decision of this paper is to determine the stacking position of each stacking unit. In this way, the scale of the problem reduces effectively. The container ship stowage problem was reasonably abstracted. Based on this, a mathematical model was constructed, whose goals are minimizing container reshuffle rate on board, minimizing the centre of gravity of the ship, holding appropriate trim and ensuring that the heavy containers stacking in the middle of the ship. Based on classification of containers, the genetic algorithm with group coding and stacking strategy in bay was taken as the resolution of the model. It can be concluded that the proposed model and the algorithm have a good performance from the analysis of the experiment's result. In this paper, four relative important factors were considered. In actual ship stowage scheduling, there are a lot of constraints. For these problems, further research is needed to improve the scheduling model.

## ACKNOWLEDGMENT

This work was supported in part by a grant from Foundation project: (1) The National Natural Science Foundation of China (70901060), (2) The Natural Science Foundation of Hubei Province (2011CDB155, 2011CDB461) (3) Open Foundation of Key Lab. for Software Engineering (SKLSE2010-08-15) (4). Supported by "the Fundamental Research Funds for the Central Universities" (5). Youth Chengguang Research Funds of Wuhan City (201150431101)

## REFERENCES

- [1] JIN Peng, HUANG Youfang, YAN Wei. Heuristic Optimization Algorithm for Reshuffle of External Container [J]. Journal of Shanghai Maritime University, 2009,12,30(4):13-16.
- [2] Zhang Yanwei, Shi Laide, Mi Weijian. Optimization Model for Reshuffle Problem of External Container in Container Yards [C]. Doctoral Student Forum of Mechanical Engineering, Shanghai, 2007:438-446.
- [3] Zhang Weiyang, Lin Yan, Ji Zhuoshang, Wu Yigang. Optimization Model of Containers Loading Operation in Export Container Terminal [J]. Journal of Wuhan University of Technology (Transportation Science & Engineering),2006,30(2):314-317.
- [4] Legato, P. Mazza, R.M. Trunfio et al. Simulation-based optimization for the quay crane scheduling problem[J]. Proceedings of the 2008 Winter Simulation Conference,2008:2717-2725
- [5] Li li, Hu Wenbin.GA-based Container Terminal Stockpile Block Plan Optimization and Simulation[C]. IEEE 2010 The Second International Conference on Networks Security, Wireless Communications and Trusted Computing, 2010:438-442.
- [6] SUN Ni, Hu Wenbin. Reserving-based Strategy for Bay Allocation [C]. IEEE The 2nd INTERNATIONAL SYMPOSIUM ON COMPUTER NETWORK AND MULTIMEDIA TECHNOLOGY,2010:291-295.
- [7] MIN Zhengyu, Hu Wenbin. Mixed-row-based Space Allocation Strategy for Outbound Containers[C].IEEE The 2nd International Conference on Information Engineering and Computer Science, 2010:231-236.
- [8] LIU Yan-bin. Model and Optimization for Multi-type Containers Ship Stowage Problem [J]. Journal of Chinese Computer Systems, 2007,28(11):1988-1991.
- [9] FAN Tiecheng. The Application of PGA in Container Loading Optimization [C]. Proceedings of the 24th Chinese Control Conference,Guangzhou,2005,1377-1381.
- [10]WANG Qiyou. Unit Loading of Container Ship[J]. Containerization,2007,18(8):26-29.
- [11]Duan Chenghua. A Stowage Planning Model for Single Bay and Multi-port Based on Integer Programming[J]. Computer Aided Engineering,2004,13(3):9-13.
- [12]ZHANG Da-bin. Genetic Algorithm based on Group Coding for Bin-packing Problem[J]. Computer Engineering and Design,2008,29(12):3154-3156.

**Wenbin Hu:** Associate professor worked in Wuhan University since 2006. He got his doctor degree in 2004.

His main research interests are intelligent simulation and optimization, multi-agent system and swarm intelligent algorithm.

**Zhengbing Hu:** He is currently Ph.D. and associate professor in Department of Information Technology, Huazhong Normal University, Wuhan, China. His present research interests include information security and data mining.