Model and algorithm for container ship stowage planning based on bin-packing problem

ZHANG Wei-ying[1,2], LIN Yan[1], and JI Zhuo-shang [1]

1. *Ship CAD Engineering Centre, Dalian University of Technology, Dalian 116024, China*
2. *College of Marine Engineering, Dalian Fisheries University, Dalian 116023, China*

Abstract: In a general case, container ship serves many different ports on each voyage. A stowage planning for container ship made at one port must take account of the influence on subsequent ports. So the complexity of stowage planning problem increases due to its multi-ports nature. This problem is NP-hard problem. In order to reduce the computational complexity, the problem is decomposed into two sub-problems in this paper. First, container ship stowage problem (CSSP) is regarded as "packing problem", ship-bays on the board of vessel are regarded as bins, the number of slots at each bay are taken as capacities of bins, and containers with different characteristics (homogeneous containers group) are treated as items packed. At this stage, there are two objective functions, one is to minimize the number of bays packed by containers and the other is to minimize the number of overstows. Secondly, containers assigned to each bays at first stage are allocate to special slot, the objective functions are to minimize the metacentric height, heel and overstows. The taboo search heuristics algorithm are used to solve the subproblem. The main focus of this paper is on the first subproblem. A case certifies the feasibility of the model and algorithm.

Keywords: container ship; stowage; bin-packing problem; heuristics algorithm

CLC number: U695.22 **Document code**: A **Article ID**: 1671 - 9433(2005)03 - 0030 - 07

1 INTRODUCTION

Container ship stowage problem (CSSP) is NP-hard problem [1-3], the size of which depends upon the ship capacity and the containers supply and demands at each port of destinations. Even for the smallest cases, CSSP has a lot of constraints to be considered, such as the hydrostatic requirements of the vessel and placement restrictions with respect to the size, kind of cargoes and strength of the containers etc.

With the increase in number of possible stowage configuration, CSSP is combinatorial explosive even for a medium-sized container-ship, Dillingham and Perakis [4] stated that the number of possible configurations for a 2 000TEU ship is approximately 3.3×10^{5735}. For example, let x_{ij} denote that the container i is assigned to cell jth on the board of ship, if m is the number of available positions on board and n is the number of containers we want to ship, $m \times n$ variables should be defined. It is clear that with vessels carrying more than 1 000 containers, the number of the required variables is greater than 10^6. Botter and Brinati [5] had set up a complete mathematical programming model for the container ship stowage problem. For a commercial size container ship of 1 000 TEU, laying at ancher in 4 harbours, their mathematical model will have nearly 10^9 decision variables and approximately 10^6 constraints, assuming that the ship will fully loaded. Up to now, it is not expected to get an optimal solution of NP-hard problem in polynomial computation time [6].

With the increase of inputs, classical optimization algorithms are ineffective to combinatorial optimization problem due to limit of computational time. People have turned their attention to analyzing different approximation techniques, viz. heuristics algorithm which can construct solution scheme corresponding with special problem, and the solutions search space can be reduced greatly because the algorithm provides an idea for solving the complexity optimization problems. The

Received date: 2005 - 04 - 29.
Fundation Item: Supported by a Special Fund Support Item of Doctor Subject of Colleges and Universities(No. 2000014125)

focus on this paper is stowage problem of the first subproblem.

2 DESCRIPTION AND SOLVING STRATEGY OF CSSP

2.1 Description of the problem

The container ship stowage problem (CSSP) in general case can be stated as follows:

Suppose that a ship is scheduled to visit a number of ports, say P, now consider the port pth of the route. At that port a number U_p of containers are unloaded, while L_p new containers are loaded. In addition there may be a number R_p of containers that, although they have subsequent destinations, must be temporarily unloaded in order to discharge the U_p containers that have to be delivered at the current port. We need to find the allocation of all containers at each port so as to facilitate this process while satisfying loading constraints, in a cost effective way, which constitutes the CSSP. Ports efficiency and ship utilization are largely determined by ship stowage planning.

2.2 Strategy for solving the stowage-planning problem

Container stowage problem is NP-hard. It is clear that it is very difficult to solve the problem at once. Generally, large container ship can require vast amounts of container movements (loading, discharge, rearrangement), that is to say it is not possible to guarantee that an optimal solution can be found for commercial sized ships in a reasonable processing time. In order to make problem easy to be solved, the CSSP is decomposed in two sub-problems.

The first subproblem is general arrangement stowage. It mainly simulates the ideas of human planner. Containers were aggregated into homogeneous groups based on some container characteristics (such as type, length, height, weight and destination) Containers that belonge to homogeneous groups are taken as a unit to be assigned to different ship bays.

At this stage, packing algorithm is used to solve the CSSP. The ship-bays are regarded as bins with the capacity of $c_j\,(j = 1,2,\cdots)$, and the number of containers with the same features at current port are regarded as item with the size s_i. Containers that belong to same destination ports at current port may be the excess number of slots ship-bay, so we should consider the situation that the size of items is larger than the bins, here items must be split into some segments. In the condition of limiting the amount of splitting, how to pack all the items into bins and make the capacity minimum, such problem is called bin-packing problem with over-sized items.

The second sub-problem is to determine the special slots of individual containers within each bay according to a series rules, such as the heavier containers at the bottom of the ship, etc.

In this paper, we focus on the first subproblem, the second one will be discussed in another paper.

3 PACKING PROBLEM WITH OVERSIZED ITEMS AND ALGORITHM

3.1 Variable sized bin-packing (VSBP) problem and bin-packing problem with over-sized items

The variable sized bins packing problem[10] can be defined as follows:

Let $A = \{a_1 \quad a_2 \quad \cdots \quad a_n\}$, $n \in z^+$ represents a set of n items, and $s_i\,(i = 1,2,\cdots,n)$ denotes the size of the item $a_i\,(1 \leqslant i \leqslant n)$, and z^+ is a set of non-integers. Let bin $= \{l_1, l_2, \cdots, l_m\}$ be a list of m types of bins, $c_j\,(j = 1,2,\cdots,m)$ denotes the capacities of bins l_j. Let $B = \{b_1,\ b_2,\cdots,b_k\}$ denote a set of bins used by an instance, the objective function of the problem is to minimize the following function:

$$\min\ f(b_1,\ b_2,\cdots,b_k) = S(B) = \sum_{j=1}^{k} c_j\ . \quad (1)$$

Bin-packing problem with over-sized items, where some item sizes are larger than the largest size of bins, is treated as a variant of the variable-sized bin-packing in practice by splitting items into some segments.

There are four algorithms for the classic bin-packing problem, namely first fit (FF), best fit (BF), first fit decreasing (FFD) and best fit decreasing (BFD). The most widely studied among these is the best fit algorithm, in which the items are packed into bin with the smallest residual capacity large enough to accommodate it. If no such bin exists, a new bin is started and the item is placed there. BF algorithm is applied to solve the stowage problem in the paper.

3.2 Bin-packing algorithm based on binary search tree

Bin-packing is NP-hard problem, so heurist algorithm can be applied to solve the problem. Use best fit algorithm to solve bin-packing problem can get an approximate solution[11-12]. Binary search tree is used to solve the problem.

3.2.1 Binary search tree (BST)

A binary search tree, which is possible empty, is a binary tree. An nonempty one should satisfy certain structural requirements. Structural properties of BST are as follows:

1) There is a distinguished node root called the root.

2) The remaining nodes are divided into two disjoint subsets, left and right, each of which is a binary search tree. Left is called the left subtree and right is called the right subtree.

3) Each node has a key and keys are different for all nodes, so the key of each node is sole.

4) The key at each node is greater than all the keys in its left subtree and lesser than or equal to all keys in its right subtree.

If leaving out the terms that all keys are different from each node in binary tree, using the equal or greater to replace the greater in term and the equal or lesser to replace the lesser in term 4), we will get a binary search tree with duplicates (DBSTree)

3.2.2 Basic ideas of using DBSTree to solve bin-packing problem

When using DBSTree solves bin-packing problem, each node of the DBSTree represents a bin that can be used to load items. Fig. 1 illustrates the following description. There are nine bins with capacity 1, 3, 12, 6, 8, 1, 20, 6 and 5, and we can use a bin search tree to store the nine bins. The capacity of each bin is the key of the nodes, so the tree is a DBSTree. The structure of the DBSTrees is shown in Fig. 1. Each value inner node presents capacity of bin, the letters outside the nodes are the name of the bins, viz. ID of each bin. If item a_i occupied four cells, viz. $S_i = 4$, to search for this key, we begin at the root and follow the left or right branch depending on whether the key sought is lesser than or greater than the key at the current node.

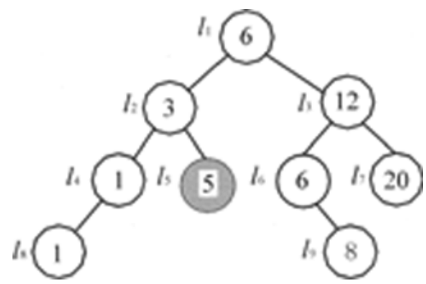

Fig. 1 Binary search tree with duplicates

As shown in Fig. 1, the key of root node represents the capacity of bin l_1, the item $a_i(s_i = 4)$ is lesser than the capacity of l_1, and so the l_1 is an alternative. According to DBSTree's characters, as the keys of the right subtree are 6 at least, the search can only be carried out in the left subtree. The bin l_2 cannot pack the item due to insufficient space. Search is transferred into right subtree of node l_2. The bin l_5 has the capacity of packing the item a_i, which becomes a rather alternative. Go on searching left subtree of node l_5 and can't find a better solution than the current because of empty left subtree of node l_5, so bin l_5 (shown in gray in Fig. 1) is the best solution for item a_i.

After finding a bin as solution, the bin should be deleted from the tree, then update the capacity of the bin by cutting its space with the size of item and insert the new to the tree unless the surplus capacity of the bin equals to zero. If no bin be found, open a new one. The above example demonstrates the general approach of solving the bin-packing problem.

3.2.3 The algorithm of best fit based DBSTree for bin-packing problem

We can use DBSTree to implement the abovementioned idea.

Step 1: Initializing capacities of all bins, index of each bin and size of all the items.

Step 2: Constructing binary search trees with duplicates based on capacity of bins and signings.

Step 3: Packing items one by one. If the size of the item is bigger than the current bin with biggest capacity, then split the item into two parts, the first part is packed into bin, which bin will be close immediately. If the remainder of the item is still bigger than the current open bins, then continue splitting until the remainder less than the bins. The remainder and the other items, whose size is not bigger than the bins, are packed according the following steps.

Step 4: Searching: viz. search the best fitting bin. Suppose the item a_i occupies room s_i, search for the key, which is large or equal to the s_i. The search starts from the root node, If the root is empty, the tree doesn't contain any key, the search fails. Otherwise compares s_i with the key of root node, if the s_i is lesser than the key, then go to the left subtree and try again. If it is greater than the value of the key, try the right subtree; if it is the same, obviously the search can be discontinued. Searches in the subtree are similar to process of above-mentioned.

After the item is packed, if the bin is full, go to step 6, otherwise go to step 5.

Step 5: Inserting node. If the bin is not full, update the capacity of the bin with remainder room. Suppose that the surplus capacity of bin is e, i. e. nodt t. The key of the nodt t inserted is compared to the value of root node, if it is lesser than that value, the left child is tried, and otherwise the right one is tested. If the node to be tested is empty, the scanning is discontinued and the nodt t becomes the child of himself. Go to step 7.

Step 6: Deleting node. If the bin is full, then the node will be deleted, there are three cases of deleting a node from the binary search tree:

Case 1: The node t to be deleted is a leaf, it has no children, and the node is disposed only.

Case 2: The node t has one child. If node t has no parent node, then abandon t, the only subtree becomes the root node of new bin search tree. If the node t has parent node *mode-f*, the parent's pointer to the node is reset to point to the node's child.

Case 3: The node has two children. The key of node is replaced by the greatest key in its left subtree or by the least key in its right subtree.

Step 7: Let all the items packed, yes go to Step 8, otherwise go to Step 3.

Step 8: Output the total number of bin used and the result of packing.

End.

4 THE MODEL AND ALGORITHM FOR CSSP

Containers to be loaded are classified according to type, destinations and sizes, which are called homogeneous containers group (HCG). According to the solving strategy of container ship calling multi-ports stowage problem, at the first stage, CSSP is identified with bin packing problem. viz. bays in the ship are regarded as bins with the capacity c_j. HCGs are regarded as item with the size s_i (the value of s_i equals to the number of containers in HCG) to be loaded into the bins. Different kind groups (sorted according to the destination, size, and type of freights) are assigned to different ship-bays, and then come into being the general arrangement stowage planning, which does not show the material location of each container, just assign the different HCGs to different ship-bays.

4.1 Basic hypotheses and defining the matrix of loading

4.1.1 Basic hypotheses

The simplifications made are presented in the following:

1) The number of container ship calling port is P, the ship starts its service route at port 1 with its all bays empty, and it sequentially visits ports 2, 3, $\cdots P$. In the last port p, all containers are discharged.

2) Suppose all containers can be loaded at each port.

3) Assume that all containers are of the same standard size with 20 ft. one 40 ft container equals to two 20 ft ones, and no hazardous and refrigerated cargo containers are considered.

4) The loading and unloading schedule is known before the beginning of the trip.

4.1.2 Matrix of transportation

Let $$\boldsymbol{T} = \begin{bmatrix} T_{12} & T_{13} & \cdots & T_{1N} \\ T_{22} & T_{23} & \cdots & T_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ T_{N2} & T_{N3} & \cdots & T_{N-1,N} \end{bmatrix} = \begin{bmatrix} T_{12} & T_{13} & \cdots & T_{1N} \\ 0 & T_{23} & \cdots & T_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & T_{N-1,N} \end{bmatrix}$$

represent container numbers at different original ports. Where T_{ij} represents the number that containers at original port i transport to destination j, when $i \geqslant j$, $T_{ij} = 0$, so the matrix is an upper triangle.

4.1.3 Homogeneous containers group (HCG)

Homogeneous containers groups are such set of containers with the same original and destination port, same type and size.

4.2 Model for CSSP

For the stowage planning of general arrangement of container ship, the following goals are considered primarily:

1) The number of bays occupied by containers at each port is as least as possible, this object ensures that HCG allocate together as possible.

2) Reducing the number of overstow operations as much as possible.

4.2.1 Parameters and design variables

Let BayID = $\{1, 2, \cdots, n\}$ denote the set of marks of the ship-bays, and Bay = $\{c_1, c_2, \cdots, c_n\}$ be set of the capacity of the BayID(i), where c_i denotes the number of cells in BayID(i) ship-bay. Group(m) = $\{s_{p1}, s_{p1}, \cdots, s_{pj}\}$ ($p = 1, 2, \cdots P$, $j = 1, 2, \cdots, P$, $m = 1, 2, \cdots, M$), define as homogeneous containers groups. s_{pj} is loading containers number from original ports p to the destinations j. Where P equals to the ports number of the ship calling, M is the number of the HCGs at current port.

$$x_{pim} = \begin{cases} 1, \text{ if group - } m \text{ container allocated} \\ \quad \text{to } BayID(i) \text{ at port } p, \\ 0, \text{ otherwise}. \end{cases} \tag{2}$$

$$y = \begin{cases} 0, & \text{case1}, \\ N_j, & \text{case2}. \end{cases} \tag{3}$$

Where y denotes the number of overstows when group s_{pj} is allocated to BayID(i). There are two cases:

Case 1: All containers on BayID(i) belong to the same destination or when mixed loading of different destination containers, destinations of the upper containers are nearer than the bellows.

Case 2: Containers at BayID(i) are more than one destinations ports and N_j is the number that the far are on top of the near.

4.2.2 Objective function

$$\min \quad \sum_{p=1}^{P}\sum_{i=1}^{n}\sum_{m=1}^{M} x_{pim}, \tag{4}$$

$$\sum_{p=1}^{P}\sum_{i=1}^{n}\sum_{m=1}^{M} x_{pim}y. \tag{5}$$

4.2.3 Constraints

$$\text{s.t.} \quad \sum_{p=1}^{P}\sum_{j=2}^{P} T_{pj} = \sum_{p=1}^{P}\sum_{j=1}^{P} s_{pj}, \tag{6}$$

$$\sum_{p=1}^{P-1}\sum_{p=1}^{j} T_{pj} - \sum_{p=1}^{P-1}\sum_{i=j+1}^{p} T_{pj} \geqslant 0, \tag{7}$$

$$\sum_{l_1}^{l_m} b_{li} \leqslant \sum_{i=1}^{n} \text{BayID}(i). \tag{8}$$

Objective function (4) expresses that the bays occupied by the same destination are as least as possible, and (5) minimizes the shifting number. In this formulation, constraints (6) is balance constraint that the containers totaled with the same kind of groups at each port equal to whole loading along the route ship called. Constraints (7) presents the number that unloading containers is greater or equals to the loading one, which can ensure all containers be loaded at a certain calling port. Constraint (8) expresses that the number of bin used when loading program is called is less or equal to the number of bay on the board of ship.

4.3 Heuristics algorithm for CSSP based on bin-packing problem with over-sized items

Step 1: Using random number initialize packing room, Bay = $\{c_1, c_2, \cdots, c_m\}$ denotes total cells at each bay on the ship, labels form fore to aft according to bay

ID viz. BayID = $\{1,2,\cdots,n\}$, or directly evaluate for each bay.

Using random number to produce matrix of transportation T_{ij}, which satisfies the term (5), a transportation matrix is said to be feasible if all the containers to be shipped can be stowed in the given bays. Use the following terms to bring feasible matrix of transportation

$$S_1 = \text{AllSlotNumber} - \sum_{j=2}^{P} T_{1j}, \tag{9}$$

$$S_p = S_{P-1} + \sum_{k=1}^{P-1} T_{kp} - \sum_{j=p+1}^{P} T_{pj}, \quad p = 2,3,\cdots,(p-1), \tag{10}$$

$$S_p \geqslant 0, \quad p = 1,2,\cdots,P, \tag{11}$$

$$\text{AllSlotNumber} = \sum_{i=1}^{n} c_i. \tag{12}$$

Where the symbol *AllSlotNumber* denotes the total cells and S_p is the number of remainder empty cells on the board of ship. Matrix of transportation satisfying formula (11) ensures all containers are loaded, and formula (12) shows that all slots on the ship equal to the summation of slot in each bay.

Step 2: Stowage for the first port

Containers are divided into P groups according to the destinations, $p = 1,2,\cdots,P$, P is the number of ship calling at.

At current calling port of the ship, sorted degressively HCGs according to destinations, HCG at the furthest destination is the first. The rest may be deduced by analogy. The HCGs are regarded as items to be packed into bins.

Using bin-packing algorithm to load the containers, if the number of containers at a port is beyond the total cells of bays, the HCG is divided into two parts, then consider it as two items.

Step 3: For the non-first calling ports, take the states after unloading as the initial situation of current port, update the capacity of the bins (bays), call bin-packing algorithm, and take formula (5) as evaluation function. If the value of the term (5) is not equal to zero, then the current bay doesn't suited with the containers of such destination, relocate to other bays which will minimize the value of the term (5).

Step 4: When all the containers at current port are loaded, go to step 5, otherwise go to step 3.

Step 5: Output the result.

End.

5 CASE STUDY

Suppose a ship services six ports in its routs. There are eight bays on the board of the ship, capacities of each bay presents in Table 1. Produce matrix of transportation according to the formulas (9), (10), (11) and (12).

$$\boldsymbol{T} = \begin{bmatrix} T_{12} & T_{13} & T_{14} & T_{15} & T_{16} \\ & T_{23} & T_{24} & T_{25} & T_{26} \\ & & T_{34} & T_{35} & T_{36} \\ & & & T_{45} & T_{46} \\ & & & & T_{56} \end{bmatrix} = \begin{bmatrix} 109 & 89 & 38 & 61 & 32 \\ 0 & 59 & 41 & 34 & 55 \\ 0 & 0 & 62 & 45 & 38 \\ 0 & 0 & 0 & 48 & 55 \\ 0 & 0 & 0 & 0 & 129 \end{bmatrix}$$

Table 1 Results of general arrangement stowage for container ship based on over-sized items bin-packing algorithm

Status	Bay 1 40	Bay 2 35	Bay 3 45	Bay 4 50	Bay 5 65	Bay6 65	Bay7 76	Bay 8 76
After loading of Port 1	38(1,4)	32(1,6)	33(1,2)	13(1,3)	—	61(1,5)	76(1,2)	76(1,3)
After unloading of Port 2	38(1,4)	32(1,6)	—	13(1,3)	—	61(1,5)	—	76(1,3)
After loading of Port 2	38(1,4)	32(1,6)	34(2,5)	13(1,3) +30(2,3)	55(2,6)	61(1,5)	41(2.4) +29(2,3)	76(1,3)
After unloading of Port 3	38(1,4)	32(1,6)	34(2,5)	—	55(2,6)	61(1,5)	41(2.4)	—
After loading of Port 3	38(1,4)	32(1,6)	34(2,5)	38(3,6)	55(2,6)	61(1,5)	41(2.4) +35(3,4)	45(3,5) +27(3,4)
After unloading of Port 4	—	32(1,6)	34(2,5)	38(3,6)	55(2,6)	61(1,5)	—	45(3,5)
After loading of Port 4	40(4,5)	32(1,6)	34(2,5)	38(3,6)	55(2,6)	61(1,5)	55(4,6) +8(4,5)	45(3,5)
After unloading of Port 5	—	32(1,6)	—	38(3,6)	55(2.6)	—	55(4,6)	—
After loading of Port 6	—	32(1,6)	—	38(3,6)	55(2,6)	53(5,6)	55(4,6)	76(5,6)

Notes: $A(B,C)$ denotes that the number of containers is A from original ports B to destination C. $A_1(B_1,C_1)+A_2(B_2,C_2)$ denotes the bay loading containers come from different original ports and to different destinations, and $A_1(B_1,C_1)$ are loaded firstly (location the underside), $A_2(B_2,C_2)$ are loaded after $A_1(B_1,C_1)$ and $A_2(B_2,C_2)$ is on top of $A_1(B_1,C_1)$.

The simulated results are in Table 1. Table 1 shows that the number of bay occupied by containers is eight and re-stows are zero when the ship calls six ports and total number of loading containers is 895TEU.

6 CONCLUSIONS

Container ship stowage problem (CSSP) is combinatorial optimization problem with multiobjects and multiconstraints. In order to reduce computational difficulty, CSSP is decomposed two subproblems in this paper, and the first one is discussed detailed. CSSP is regarded as bin-packing problem. After being modeled, the algorithm of best fit for bin-packing problem is used to solve the CSSP. A case shows that the algorithm is feasible for container ship stowage problem.

REFERENCES

[1] AVRIEL M, PENN M, SHPIRER N, WITTEBON S. Stowage planning for container ships to reduce the number of shifts [J]. Annals of Operations Research, 1998, 76:55 -71.

[2] WILSON I D, ROACH P A. Principles combinatorial optimization applied to container-ship stowage planning [J]. Journal of Heuristics, 1999(5):403 -418.

[3] AVRIEL M, PENN M, SHPIRER N. Container ship stowage problem: complexity and connection to the coloring of circle graphs [J]. Discrete Applied Mathematics, 2000, 103: 271 -279.

[4] DILLINGHAM J T, PERAKIS A N. Application of artificial intelligence in the marine industry [A]. Fleet Management Technology Conference[D]. Boston, 1986.

[5] BOTTER R C, BRINATI M A. Stowage container planning: a model for getting an optimal solution [A]. ICCAS'92[C]. North Holland, 1992: 217 -229.

[6] XIE Jinxing, XING Wenxun. Modern optimization algorithm [M]. Beijing: Tsinghua University Press, 2000(in Chinese).

[7] JONATHAN J. SHIELDS. Container ship stowage: a computer-aided preplanning system [J]. Marine Technology, 1984, 21:370 -383.

[8] KEMP S E, ROACH P A, WARE A J, WILSON J D. Artificial intelligence for automatic container stowage planning optimization [J]. HANSA-Schiffahrt-Schiffbau-Hafen -141. 2004(4): 22 -26.

[9] SARTAJ S. Data structures, algorithms, and applications in C + + [M]. Beijing: China Machine Press, 1999(in Chinese).

[10] XING Wenxun. A bin-packing problem with over-sized items [J]. Operations Research Letters, 2000, 30: 83 -88.

[11] GAREY M, GRAHAM K, JOHNSON D, YAO A. Resource constrained scheduling as generalized bin-packing [J]. Journal of Combinatorial Theory, 1976, Series A: 257 -298.

[12] JOHNSON D, DEMERS A, ULLMAN J, GAREY M, GRAHAM R. Worst-case performance bounds for simple one-dimensional packing algorithms [J]. SIAM Journal on Computing, 1974: 299 -325.

ZHANG Wei-ying, born in 1963, received B. S. and M. S. degree in mechanical engineering from Liaoning Technical University in 1985 and 1996 respectively. Now she is a vice professor in Dalian Fishery University and is currently working for her Ph. D degree in ship CAD engineering center of Dalian University of Technology, Her research interests are fuzzy optimization of ship form and optimization of container ship stowage problem.

LIN Yan, professor, doctoral supervison, was born in 1963. He received the B. S. and M. S. degree from Dalian university of Technology in 1984 and 1987 respectively. and then got his doctor degree in Shanghai Jiaotong University in 1993. He published many papers in ship design area. His main research fields: ship form developing and special forming technology of ship hull.