Journal of Marine Science and Technology

Volume 12 | Issue 3 Article 6

A Two-Stage Optimization of Piece Arrangement for the Cutting Problem in Shipbuilding

Tso-Chung Sung
Ph.D. Candidate, Department of System Engineering and Naval Architecture, National Taiwan Ocean University, Keelung, Taiwan 20224., tcsung@se.ntou.edu.tw

Wei-Chu Weng
Assistant Professor, Department of System Engineering and Naval Architecture, National Taiwan Ocean University, Keelung, Taiwan 20224.

Chien-Tung Yang
Associate Professor, Department of System Engineering and Naval Architecture, National Taiwan Ocean University, Keelung, Taiwan 20224.

Recommended Citation
Sung, Tso-Chung; Weng, Wei-Chu; and Yang, Chien-Tung (2004) "A Two-Stage Optimization of Piece Arrangement for the Cutting Problem in Shipbuilding," *Journal of Marine Science and Technology*: Vol. 12: Iss. 3, Article 6.
DOI: 10.51400/2709-6998.2236
Available at: https://jmstt.ntou.edu.tw/journal/vol12/iss3/6

A TWO-STAGE OPTIMIZATION OF PIECE ARRANGEMENT FOR THE CUTTING PROBLEM IN SHIPBUILDING

Tso-Chung Sung*, Wei-Chu Weng**, and Chien-Tung Yang***

Key words: piece arrangement, cutting stock problem (CSP), two-stage optimization, pre-selection and elimination technique, tabu search.

ABSTRACT

In this paper a two-stage optimization of piece arrangement for the cutting problem in shipbuilding is proposed. The two-stage optimization consists of a new technique, which is named as pre-selection and elimination technique (PSET), at the first stage and tabu search method at the second stage. Applying the new technique PSET, certain longer orders, which are chosen by a control factor, will be matched with some other shorter ones to fit the stock of shaped steel. Once these orders are selected, they will be eliminated from the whole orders. After the first stage, a tabu search approach is applied to optimize the remainder arrangement at the second stage. By using this two-stage optimization, both the solution quality and computational time are highly improved.

INTRODUCTION

A ship hull is constructed of many blocks which are different in shape and size. Each block is assembled with many pieces of shaped steel. Due to different shape and different size of blocks, the length differs considerably between pieces of shaped steel. Engineers have to prepare sufficient stocks of shaped steel to cut into these pieces of shaped steel. The work of piece arrangement is to arrange some pieces of shaped steel to each selected stock of shaped steel until all pieces of shaped steel have been assigned. The problem discussed in this paper is to minimize the total wasted length while all pieces of shaped steel are cut from the stocks of shaped steel with constant length.

According to the definition in reference (Dyckhoff, 1990), piece arrangement is an one-dimensional general cutting stock problem (CSP) where the stocks of shaped steel are termed as stocks, the pieces of shaped steel are termed as orders and the total wasted length is considered as total trim loss. CSP is NP-complete and a solution can be found mostly by using approximate methods and heuristics (Gradisar and Trkman, 2004). In the past, CSP was solved by method of linear programming (Glimore and Gomory, 1961, 1963). A hybrid method was proposed for this problem (Scholl *et al*., 1997). Genetic algorithm (GA) has also been applied to such problem successfully (Kos and Duhovnik, 2000). However, there are only few researches for shipbuilding industry. In reference (Weng and Hung, 2003), CSP was discussed and approached by GA especially for shipbuilding industry.

When compared with the most cutting stock problems existing in timber industry and aluminum industry, ordered lengths demanded by shipbuilding industry contain more types of length and lower plurality in where some of lengths are very close to the longest stock length. By using the multiple types of stock length, the work of piece arrangement can be carried easier than by using one constant stock length. However, the administrative staff of the storage will have the problems of complexities to handle the multiple types of stock length. These kinds of problem can be solved by using the stocks with constant length. In this paper, only the constant stock length is used.

In the past, tabu search (TS) has been successfully applied to sequencing problems, such as scheduling problems (Dorn *et al*., 1998; Marett and Wright, 1996; Murata and Ishibuchi, 1994; Weng, 2002; Weng *et al*., 2003), and routing problem (Weng and Hung, 2002). The method of tabu search implements the search from one local area to another local area in the solution space. A local area is a point set of solutions where each one is generated from a specific point, termed as a reference

Paper Submitted 04/14/04, Accepted 07/02/04. Author for Correspondence: Tso-Chung Sung. E-mail: tcsung@se.ntou.edu.tw.
**Ph.D. Candidate, Department of System Engineering and Naval Architecture, National Taiwan Ocean University, Keelung, Taiwan 20224.*
***Assistant Professor, Department of System Engineering and Naval Architecture, National Taiwan Ocean University, Keelung, Taiwan 20224.*
****Associate Professor, Department of System Engineering and Naval Architecture, National Taiwan Ocean University, Keelung, Taiwan 20224.*

point, by a predefined move function. The generated local area is termed as the neighborhood of this reference point. A reference point is generated randomly at the beginning and then will be iterated by the best point searched in current neighborhood for generating next neighborhood. The best reference point ever found is the result after tabu search has been stopped. During the search, a list of old reference points, termed as the tabu list, exists to prevent cyclic search.

According to our experiences, the traditional tabu search algorithm will face the problem that the searching time increases very fast when the problem size increases. In order to overcome this problem, several combined or hybrid methods were proposed. A combined method of sequence heuristic procedure (SHP) and branch-and-bound was introduced by Gradisar and Trkman (2004) for the purposes of lower trim loss and acceptable time complexity. A hybrid method which consists of reduction and fit techniques associated with a dualtabu strategy was proposed by Scholl *et al*. (1997) for the purposes of increasing the solution quality and reducing the computational time. In this paper, a two-stage optimization scheme, where a pre-selection and elimination technique (PSET) is proposed for the first stage and tabu search is adopted for the second stage, is designed to improve this situation. Two real cases for the piece arrangement of ship construction are examined to demonstrate the efficiency and effectiveness of this two-stage optimization.

PROBLEM STATEMENT

Under the assumption of modeling the problem in this paper as a problem of deciding the cutting sequence, there are total n orders in set **P**, items with same length are treated as different ones, where the length of order j is l_j and $1 \le j \le n$. It is assumed that at least m stocks with given length L to be cut into all orders by following a given cutting sequence $\boldsymbol{s}$. Cutting sequence $\boldsymbol{s}$ is a sequence of all items $\{s_1, s_2, s_3, ..., s_i, ..., s_n\}$ where s_i represents the item number of the ith element. The optimization of cutting sequence problem can be defined as to find an optimal $\boldsymbol{s}$ where the objective function is:

$$\text{To minimize} \quad T = \{\sum_{i=1}^{m} t_i\} \tag{1}$$

$$t_i = L - \sum_{j=k1_i}^{j=k2_i} l_{s_j} \quad 1 \le k1_i \le k2_i \le n \tag{2}$$

where m :the number of stocks used

t_i :the trim loss of the ith stock

T :the total trim loss

$k1_i$:the $k1_i$ th element in $\boldsymbol{s}$ which is the first order assigned to the ith stock

$k2_i$:the $k2_i$ th element in $\boldsymbol{s}$ which is the last order assigned to the ith stock

The constraint of this optimization problem is that all orders must be assigned to stocks. A trim loss ratio ρ is introduced for comparison and defined as follows:

$$\rho = \frac{T}{L_e} \% \tag{3}$$

$$L_e = \sum_{j=1}^{n} l_i \tag{4}$$

Where, L_e is the total length of orders. In Figure 1, a small example of cutting sequence $\boldsymbol{s} = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$ is introduced to show the relationship between the cutting sequence and piece arrangement where the gray bar represents the respective trim loss.

PRE-SELECTION AND ELIMINATION TECHNIQUE

The main goal of the new proposed pre-selection and elimination technique (PSET) is to generate a partial cutting sequence $\boldsymbol{s}_{PSET}$ and then to reduce the size of problem that the tabu search method will solve at the second stage. It is the work to match so-called lengthy orders with other paired/single short ones. The idea of matching is partially similar to the combined method of first-fit-decreasing (FFD) and best-two-fit (B2F) adopted by Scholl *et al*. (1997) for off-line bin packing problem (BPP). However unlike the using of FFD to fit the orders into stocks at the beginning, only a predefined control factor r_{PSET} is used to determine the selected lengthy orders by PSET. Each single selected lengthy order will be placed in the front of a new stock. The residual length of those stocks will be fitted by the paired-items and/or single item of the remaining unselected short orders. After this work, a partial cutting sequence will be formed. The remained items of the short part will be solved by tabu search and it has more opportunities to obtain the optimal partial cutting

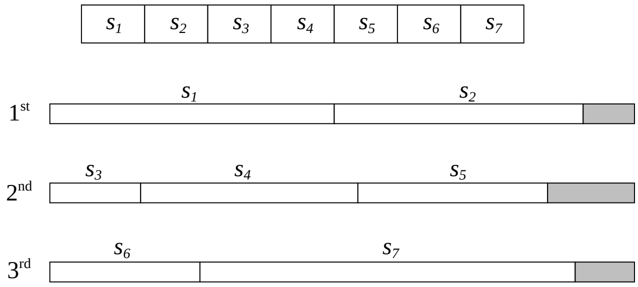

Fig. 1. Diagram of cutting sequence and piece arrangement.

sequence for the reason of reduced problem size. In view of reducing the problem size, the scheme of PSET is much more simple than the combined method of FFD + B2F. Although PSET is capable of reducing the size of problem for the optimization of next stage, it still has the restriction of using. Basically, PSET assigns four orders into a single stock at the most. Therefore, PSET is not suitable for a situation that all of the orders are much smaller than half of stock length.

For the PSET, the control factor rPSET is termed as the selection factor. Selected lengthy items $\mathbf{P}_L$ are those whose lengths are greater than or equal to $L \times r_{PSET}$. For unselected orders, a candidate pool can be generated which containing all kinds of single candidate and paired-candidate. Each single candidate is an individual unselected order itself. Any combination of two unselected items is a paired-candidate. First of all, all items of set $\mathbf{P}_L$ are sequenced according to length in ascending. Then, the candidate pool **C** is formed with all kinds of paired candidates and single ones. In addition, they are sequenced from paired candidates to single ones and then according to candidate length in descending individually.

To some lengthy item, a matched candidate is the best if total length of the lengthy item and the matched candidate makes the requested stock having minimum trim loss. Each element in $\mathbf{P}_L$ will be sequentially matched with one of candidates in candidate pool **C**. After the matching procedure, the set $\mathbf{P}_{PSET}$ can be formed with all of the orders in $\mathbf{P}_L$ and those contained in best-matched candidates. They will be eliminated from the set of order **P**. Only items in set $\mathbf{P}_2$ are going to be arranged optimally by tabu search.

For example, there is a problem of total 7 orders $\mathbf{P} = \{1, 2, 3, 4, 5, 6, 7\}$, where item 1 to item 3 are the lengthy items selected by PSET, i.e. $\mathbf{P}_L = \{1, 2, 3\}$. From the unselected orders, item 4 to item 7, total 10 kinds of candidate can be generated, which includes 4 kinds of single candidate which are item 4, item 5, item 6, and item 7 and 6 kinds of paired candidate which are (4, 5), (4, 6), (4, 7), (5, 6), (5, 7) and (6,7). Before matching, all lengthy items will be sequenced according to length in ascending, e.g. $\mathbf{P}_L = \{3, 1, 2\}$ for instance. In addition, all candidates will be permutated from the paired one to the single one and then according to the total length in descending individually, e.g. $\boldsymbol{c} = \{(5, 7), (4, 5), (4, 7), (5, 6), (6, 7), (4, 6), (5), (7), (4), (6)\}$ for instance. The permutated lengthy items are going to be matched with permutated candidates one by one. For example, for the selected order of item 3, candidate (4, 7) is assumed to be the best one if the trim loss of assigned stock will be the minimum when compared with other candidates. Then, each candidate containing item 4 or item 7 is disappeared automatically. The rest permutated candidates are {(5,6), (5), (6)}. If the best is candidate {(5)} when matching with the second selected item (item 1), then the rest candidate is {(6)}. The candidate of the third selected item (item 2) should be candidate {(6)}. Consequently, three stocks are assigned to cut into items of {3, 4, 7}, items of {1, 5} and items of {2, 6}, respectively, i.e. the decided cutting sequence is $\boldsymbol{s}_{PSET} = \{3, 4, 7, 1, 5, 6\}$ and $\mathbf{P}_{PSET} = \{1, 2, 3, 4, 5, 6, 7\}$ and $\mathbf{P}_2 = \varnothing$.

The pre-specified selection factor rPSET affects the result of piece arrangement. It will be examined and discussed in the example section later. The algorithm of PSET can be described as below:

Step 1: Decide the set $\mathbf{P}_L$ based on rPSET; Generate the candidate pool **C**.
Step 2: Sort elements in set $\mathbf{P}_L$ according to its length in ascending; Sort candidates in set **C** from paired one to single one and then according to the candidate's length in descending individually;
Step 3: Match each element in $\mathbf{P}_L$ sequentially with each on in **C** and find the best.
Step 4: Decide the set $\mathbf{P}_{PSET}$; $\mathbf{P}_2 = \mathbf{P} - \mathbf{P}_{PSET}$, then stop.

TABU SEARCH

Tabu search (TS) iteratively examines the solution space **X** which is a solution set including all kinds of feasible solution in a form of vector $\boldsymbol{s}_2$. Initially, TS starts the exploration from a random solution point as the first reference point $\boldsymbol{x}$. Based on this initial reference point, all other solution points can be completely generated by the predefined move function $\boldsymbol{m}$ and grouped as a neighborhood **Z**. Any point of current neighborhood found to be the best solution is termed as a local best point $\boldsymbol{x}'$. Current local best point will surely become next reference point to get into the searching procedure of next neighborhood. There exists a variable $\boldsymbol{o}_x$, termed as the overall best point, which always records the best of local best point found up to now. Current reference point will be appended to the tabu list **T** in order to be excluded from next neighborhood. This is a design to prevent cyclic search due to any reference point duplicated. The solution contained in variable $\boldsymbol{o}_x$ becomes the search result when TS stops. Once of TS implementation induces one result of searching.

1. Solution representation and move function

Each solution point is represented in a form of vector $\boldsymbol{s}_2$ containing sequenced n_2 items in set $\mathbf{P}_2$. The move function can be therefore designed as to insert one specified item to other specified location (Tallard, 1993). For any known reference point $\boldsymbol{x} = \{x_1 , x_1 , x_1, x_1, x_1, ...,$

$x_{n_2}\}$, the operation of move function $\boldsymbol{m}(\boldsymbol{x}, i, j)$ is to move the ith element behind the jth element and can be defined as below:

$$\boldsymbol{m}(\boldsymbol{x}, i, j) = \{x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{j-1},$$

$$x_j, xi, x_{j+1}, \ldots, x_{n_2}\} \tag{5}$$

For example, a point $\boldsymbol{m}(\boldsymbol{x}, 3, 7)$ in $\boldsymbol{x}$'s neighborhood can be generated as shown in Figure 2 if $\boldsymbol{x} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$. All combinations of parameter i and j control the move and insertion operation to generate all points in the $\boldsymbol{x}$'s neighborhood $\mathbf{Z}$. The neighborhood $\mathbf{Z}$ can be defined as following:

$$\mathbf{Z} = \{\boldsymbol{m}(\boldsymbol{x}, i, j) \mid 1 \le i, j \le n$$

$$\text{where } j \ne i \text{ and } j \ne i - 1 \} \tag{6}$$

If $j = i$, this is not a valid operation and should be excluded. Another excluded condition is when $j = i - 1$ where no change is going to happen to the sequence $\boldsymbol{x}$ after such operation. One more condition is when an insertion results in any change only happened locally within a partial sequence for one stock. Such operation won't affect the total trim loss and will be excluded automatically by programming. For example, as shown in figure 1, neighborhood points such as $\boldsymbol{m}(\boldsymbol{x}, 3, 4)$, $\boldsymbol{m}(\boldsymbol{x}, 3, 5)$, $\boldsymbol{m}(\boldsymbol{x}, 4, 5)$ and $\boldsymbol{m}(\boldsymbol{x}, 5, 3)$ won't affect the trim loss of second stock as well as the total trim loss.

2. Tabu list and effective neighborhood

According to move function defined above, a reference point can be used to generate one and only one neighborhood. Therefore, during the searching of TS, any reference point $\boldsymbol{x}$ repeated will result in cyclic search. The tabu list contains all old reference points to prevent this situation. Any neighborhood $\mathbf{Z}$ excluding all points in tabu list is termed as the effective neighborhood. This effective neighborhood $\mathbf{Z}_{eff}$ is the point set what TS truly searches for the local best point.

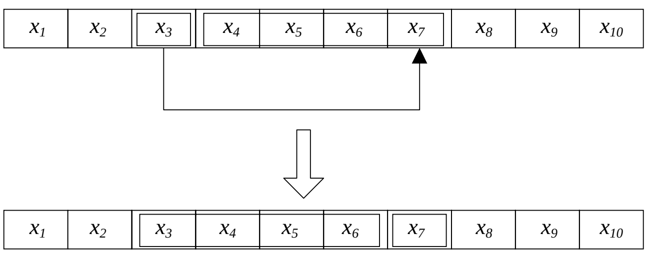

Fig. 2. The operation of move function $\boldsymbol{m}(x, 3, 7)$.

3. Stop criteria

TS won't stop the search when any local minimum or maximum is reached. It is stopped due to some critical situations. The basic situations include the following two:

(1) Current neighborhood is empty.

(2) The maximum number of total neighborhoods is reached.

These two stop criteria are used in our examples.

4. Algorithm of tabu search

Step 1: Input the number of total neighborhoods to be searched: *maxZ*;
Initialize relative parameters:
The length of tabu list: *lengthT*;
Generate a random initial solution $\boldsymbol{x}_0$;
ZCounter : = 0, $\boldsymbol{x}$: = $\boldsymbol{x}_0$ and $\mathbf{T} = \varnothing$.
$\boldsymbol{o}_{\boldsymbol{x}}$: = $\boldsymbol{x}$.
Step 2: Generate all points in current $\mathbf{Z}$;
Decide the effective neighborhood $\mathbf{Z}_{eff}$;
Step 3: If $\mathbf{Z}_{eff}$ is empty, then stop;
Otherwise
Set *ZCounter* : = *ZCounter* + 1;
Find current $\boldsymbol{x}' = optimum(\boldsymbol{z} : \boldsymbol{z} \in \mathbf{Z}_{eff})$.
Step 4: If $\boldsymbol{x}'$ is better than $\boldsymbol{o}_{\boldsymbol{x}}$ then Let $\boldsymbol{o}_{\boldsymbol{x}} := \boldsymbol{x}'$;
Append $\boldsymbol{x}'$ to $\mathbf{T}$;
$\boldsymbol{x} := \boldsymbol{x}'$.
Step 5: If *ZCounter* = *maxZ*, then stop;
Otherwise go to step 2.

TWO-STAGE OPTIMIZATION

Since both PSET and tabu search scheme have been introduced in the previous sections, a combined two-stage optimization scheme will be discussed in this section. The two-stage optimization scheme consists of PSET as the first stage scheme for the purposes of both generating a partial cutting sequence and reducing the problem size for next stage and tabu search as the second stage optimization method for the solution of another partial cutting sequence. For the first stage, selected lengthy orders $\mathbf{P}_L$ by using a control factor will be matched with some of unselected orders one by one. A partial sequence $\boldsymbol{s}_{PSET}$, the first part of sequence $\mathbf{s}$, will be generated by PSET during this stage. A set of orders $\mathbf{P}_{PSET}$ are therefore formed which includes the orders selected and those matched with. For the second stage, the orders $\mathbf{P}_2$ left by PSET, i.e. $\mathbf{P}_2 = \mathbf{P} - \mathbf{P}_{PSET}$, are those whom tabu search is chosen to arrange optimally. In other words, the set of order $\mathbf{P}$ is divided into two parts where PSET handles the set $\mathbf{P}_{PSET}$ including n_{PSET} orders and the rest in set $\mathbf{P}_2$ for the second stage including $\boldsymbol{n}_2$

pieces where $n = n_{PSET} + n_2$. The partial sequence $\boldsymbol{s}_2$, the second part of sequence s, is decided according to the optimization result of tabu search. Each partial cutting sequence, $\boldsymbol{s}_{PSET}$ or $\boldsymbol{s}_2$, has its own requested stocks to cut. Whole algorithm is described as a flowchart shown in Figure 3.

EXAMPLES

In this section, two real cases for ship hull construction are introduced as examples in this paper. In the first case, 147 orders across twelve ship construction blocks are to be determined for piece arrangement.

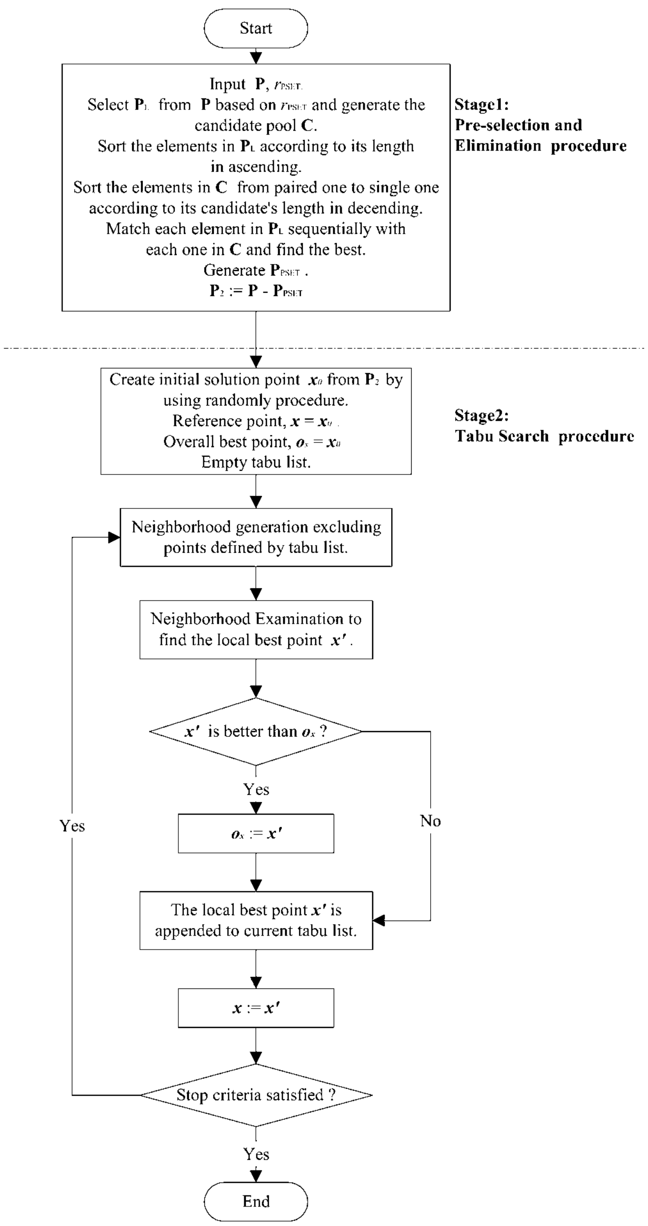

Fig. 3. The flowchart of two-stage optimization.

For the second case, 43 orders are presented. Practically, there are 11 kinds of standard stock length used by shipbuilding companies in Taiwan as listed in Table 1, while only two of the longest ones are feasible for the longest order shown in these two cases. Therefore, two conditions will be examined for each case where condition A is the stock length L = 15,000 mm and condition B is L = 14,900 mm. Two kinds of situation when r_{PSET} = 0.75 and r_{PSET} = 0.5 are tested individually for each case under each condition respectively.

Tabu search explores the solution space from an initial solution point and stops when the preset amount of total searched neighborhoods is reached. Different initial solution point induces different result. In addition, according to our experiences, the result of TS is hardly improved after around 10 to 20 neighborhoods have been searched. In order to search results as good as possible, multiple examinations with different initial solution is usually adopted. Therefore, the result of TS respective to different r_{PSET} for each case under each condition shows the best result over 50 times of examination each of which searches total 25 neighborhoods. In other words, tabu search will examine the solution space 50 times each of which is started from different initial solution and stopped after 25 neighborhoods have been searched. All computations are programmed in FORTRAN and carried out on the PC with CPU Pentium IV/2.4 Ghz and RAM 512 MBs.

Table 2 and Table 3 show numerical results of case 1 under condition A and condition B, respectively, while Table 4 and Table 5 show those of case 2 under condition A and condition B, respectively. As shown in each table, respective to the value of r_{PSET}, each row lists the amount of total number of orders processed by PSET and TS respectively and the total trim loss ratio. Within all tables, the result of TS is referred to reference (Sung *et al.*, 2004) and shown for comparison, which were the best result over 50 times of examination with different initial solution and implemented by searching 25 neighborhoods.

The computation time taken by two-stage optimization for each case under each condition is shown in Table 1. Most of time taken by two-stage optimization is consumed by TS on second stage while PSET took at most 3.89 seconds for case 1 and at most 0.04 seconds for case 2. The time taken when only TS is applied to process all items is also shown in Table 6 for comparison.

As shown in each of Table 2 to Table 5, the total trim loss ratio ρ decreases apparently when r_{PSET} decreases. Lower selection factor r_{PSET} represents that more orders are selected due to looser definition of lengthy item. On the first stage, paired candidates were enumerated to offer different matching choices that may be better than single ones. In addition, PSET matched selected items with the best candidate and solved the partial sequence $\boldsymbol{s}_{PSET}$ as good as possible. On second stage, the solution space is shrunk widely when selection factor r_{PSET} is lowered down. There will be more opportunity to find the global optimum solution within a smaller solution space.

For CSP with constant stock length, the commonly used objective is to minimize the amount of requested stocks. The minimum number of stocks required m_{LB} under this objective is 33 stocks for case 1 and 14 stocks for case 2 with both condition A and condition B. The m_{LB} can be calculated in the way that total length of orders L_e divided by single stock length L. If decimal part exists in the result of calculating, the m_{LB} should be added one. As can be seen in Table 2 to Table 5, by two-stage optimization, the global minimum amount of requested stocks has been reached when r_{PSET} = 0.5 for case 1 under condition A and all values of r_{PSET} for case 2 under both conditions. For case 2, the total trim loss ratio under condition B is lower than that under condition A due to stocks with smaller length given.

The selection factor r_{PSET} decides how two heuristic methods work together and affects the result and efficiency. High selection factor may cause that, on second stage, TS will take much time to explore the solution space not so small that unsatisfying results will be found. Low selection factor may cause that, for lengthy items with short lengths, PSET will take much time in enumerating other candidates composed of three or more items and in matching with the best candidate.

As shown in Table 6, when two-stage optimization is applied, the computation time of TS has apparently been lowered down up to 60% for case 1 and 92% for case 2. The computation time taken on first stage (PSET) is tiny when compared with that taken on second stage (TS). On the other hand, on second stage, TS takes

Table 1. The list of available types of stock length usually used in the shipbuilding industries in Taiwan

Type No.	Length (mm)	Type No.	Length (mm)	Type No.	Length (mm)
1	5,500	5	11,000	9	14000
2	8,500	6	11,500	10	14900
3	9,000	7	12,500	11	15000
4	10,500	8	13,000		

Table 2. Numerical results of case 1 under condition A (L_e = 483,559 mm)

Method	Amount of processed items			ρ
	PSET	TS	Total	
TS [10]	0	147	147	5.460%
Two-stage opt. (r_{PSET} = 0.75)	17	130	147	5.460%
Two-stage opt. (r_{PSET} = 0.50)	38	109	147	2.366%

Table 3. Numerical results of case 1 under condition B (L_e = 483,559 mm)

Method	Amount of processed items			ρ
	PSET	TS	Total	
TS [10]	0	147	147	4.765%
Two-stage opt. (r_{PSET} = 0.75)	17	130	147	4.765%
Two-stage opt. (r_{PSET} = 0.50)	38	109	147	4.765%

Table 4. Numerical results of case 2 under condition A (L_e = 201,870 mm)

Method	Amount of processed items			ρ
	PSET	TS	Total	
TS [10]	0	43	43	4.027%
Two-stage opt. (r_{PSET} = 0.75)	13	30	43	4.027%
Two-stage opt. (r_{PSET} = 0.50)	18	25	43	4.027%

Table 5. Numerical results of case 2 under condition B (L_e = 201,870 mm)

Method	Amount of processed items			ρ
	PSET	TS	Total	
TS [10]	0	43	43	10.715%
Two-stage opt. (r_{PSET} = 0.75)	13	30	43	3.334%
Two-stage opt. (r_{PSET} = 0.50)	18	25	43	3.334%

Table 6. The computation time (measured in seconds)

Method	Case 1		Case 2	
	Cond. A	Cond. B	Cond. A	Cond. B
TS [10]	378	375	16	15.5
Two-stage opt. (r_{PSET} = 0.75)	251	250	5.5	1.3
Two-stage opt. (r_{PSET} = 0.50)	152	155	5.5	1.3

most time to evaluate the value of objective function for each solution point in each neighborhood. When selection factor r_{PSET} was lowered down, the time taken by PSET was increased within limits. On the other hand, the time taken by TS was therefore decreased apparently because the amount of solution point in each neighborhood was decreased widely and the time to evaluate the value of objective function for each solution point is decreased due to small amount of processed items.

CONCLUSION

A two-stage optimization scheme, which combined PSET and tabu search method, is proposed for the piece arrangement of ship hull construction. PSET, the first stage of the proposed technique formed by a selection, fit and elimination scheme, has successfully reduced the size of the problem that the tabu search method will solve at the second stage only by using of the control factor r_{PSET} appropriately. Two real cases taken from the Keelung shipyard of China Shipbuilding Corporation in Taiwan have been solved by the proposed two-stage optimization scheme. For the case 1, when r_{PSET} is equal to 0.75 and 0.5 separately, the number of the orders processed by PSET is 11.6% and 25.9% of the total number of the orders, respectively. For the case 2, under the same settlement situation of r_{PSET}, the percentage of the number of the orders processed by PSET is 30.2% and 41.9%, separately. It is obvious that PSET can reduce problem size by using the control factor r_{PSET} appropriately. Besides, it can be observed clearly that the PSET has handled the most the lengthy orders with short ones. These so-called lengthy orders, which length are almost close to or over half of the longest stock length, are used commonly in the ship hull construction of shipyard. Therefore, the proposed PSET scheme is a proper method for the work of piece arrangement in the shipyard. During the processes of the proposed two-stage optimization, the control factor plays an important role of dividing the orders into two parts. The determination of the control factor relies on the assortment of both length and quantity of orders that wait to be arranged. It is our further research works in the future.

REFERENCES

1. Dorn, J., Girsch, M., Skele, G., and Slany, W., "Comparison of Iterative Improvement Techniques for Schedule Optimization," *Euro. J. Operat. Res.*, Vol. 94, pp. 349-361 (1998).
2. Dyckhoff, H., "A Typology of Cutting and Packing Problems," *Eur. J. Operat. Res.*, Vol. 44, pp. 145-159 (1990).
3. Glimore, P.C. and Gomory, R.E., "A Linear Programming Approach to the Cutting Stock Problem," *Operat. Res.*, Vol. 9, pp. 849-859 (1961).
4. Glimore, P.C. and Gomory, R.E., "A Linear Programming Approach to the Cutting Stock Problem: Part II," *Operat. Res.*, Vol. 11, pp. 863-888 (1963).
5. Gradisar, M. and Trkman, P., "A Combined Approach to the Solution to the General One-Dimensional Cutting Stock Problem," *Computers Operat. Res.*, In Press (2004).
6. Kos, L. and Duhovnik, J., "Rod Cutting Optimization with Store Utilization," *International Design Conference – Design 2000*, May 23-26, Dubrovnic, pp. 313-318 (2000).
7. Marett, R. and Wright, M., "A Comparison of Neighborhood Search Techniques for Multi-Objective Combinatorial Problems," *Computers Operat. Res.*, Vol. 23, No. 5, pp. 465-483 (1996).
8. Murata, T. and Ishibuchi, H., "Performance Evaluation of Genetic Algorithms for Flowshop Scheduling Problems," *Evolution Computation, IEEE World Congress on Computational Intelligence, Proceedings of the First IEEE Conference*, Vol. 2, pp. 812-817 (1994).
9. Scholl, A., Klein, R., and Jurgens, C., "BISON: A Fast Hybird Procedure for Exactly Solving the One-Dimensional Bin Packing Problem," *Computers Operat. Res.*, Vol. 24, No. 7, pp. 627-645 (1997).
10. Sung, T.C., Weng, W.C., and Yang, C.T., "Optimization of Piece Arrangement for One-Size Source Cutting Problem by Tabu Search for Shipyard," *The Sixteenth National Conference on Naval Architecture and Marine Engineering*, Tainan, Taiwan, March 5-6 (2004). (in Chinese)
11. Tallard, E., "Benchmarks for Basic Scheduling Problems," *Euro. J. Operat. Res.*, Vol. 64, pp. 278-285 (1993).
12. Weng, W.C., "The Application of Tabu Search to Combinatorial Problems," Ph.D. Dissertation, Department of Engineering Science and Ocean Engineering, National Taiwan University, Taipei, Taiwan (2002).
13. Weng, W.C. and Hung, C.F., "Optimization of Working

Process for Ship Structures by Tabu Search Method," *Bull. College Engin., Nat. Taiwan Univ., Taipei, Taiwan*, Vol. 86, pp. 61-72 (2002). (in Chinese)

14. Weng, W.C. and Hung, C.F., "The Optimization of a Two-stage FSPM Under Space Constrain by Tabu Search," *J. Taiwan Soc. Naval Architect. Marine Engin.*, Vol. 22, No. 3, pp. 133-141 (2003).
15. Weng, W.C., Yang, C.T., and Hung, C.F., "The Optimization of Section Steel Arrangement for Ship Construction Associated with Cutting Rule by Genetic Algorithm," *The Seventeenth Asian-Pacific Technical Exchange and Advisory Meeting on Marine Structure, National Cheng Kung University*, Tainan, Taiwan, pp. 237-246 (2003).