

Research Article

Spatial Scheduling Optimization Algorithm for Block Assembly in Shipbuilding

Zhengyang Shang, Jinan Gu, Wei Ding, and Enock A. Duodu

Mechanical Information Research Center, Jiangsu University, Zhenjiang 212013, China

Correspondence should be addressed to Zhengyang Shang; shzy0205@163.com

Received 21 February 2017; Accepted 9 March 2017; Published 18 April 2017

Academic Editor: Paolo Boscariol

Copyright © 2017 Zhengyang Shang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Block assembly consumes the majority of processing time and resources in shipbuilding, and the block spatial scheduling (BSS) related to block assembly has been widely studied as the key to improve shipbuilding efficiency. BSS is a complicated NP-hard problem that aims to minimize the makespan. Since each block has specific building time and space constraints, the BSS problem can be hardly found with an acceptable solution by using constant scheduling rules. Thus, in this study, we considered the BSS problem as a time-constrained 3D bin packing mathematical model and proposed an allocation algorithm, best contact algorithm (BCA), that is more suitable for dynamic processes. Then, for global optimization of the BSS problem, we regarded the starting time of each block as a variable and used the genetic algorithm (GA) to operate and optimize the block assembly sequence. Finally, we tested the BCA + GA scheduling system with real data from a shipyard and thereby determined the block scheduling status and the daily utilization rate of the work plate. Comparison shows that the proposed algorithm is able to get shorter makespan and better block scheduling effect; it realized the optimization of the block spatial scheduling dynamically.

1. Introduction

Shipbuilding is a traditional mechanical manufacturing process that requires enough building space and resources due to its large and complex components. In the construction process, the components are built and transported as a basic element, called block. It is an intermediate product of dispatch and process in modern shipbuilding. Shipbuilding generally involves six stages: steel material processing, block assembly, outfitting, hull erection, commissioning, and trial trip. Among them, block assembly consumes the maximum building time and thus is considered as a bottleneck in improving production efficiency and reducing the makespan in the shipyard. Therefore, this study is focused on reasonable block scheduling in the assembly process.

Block-based shipbuilding is very complicated and requires the joint action of diverse resources, such as space, manpower, equipment, overhang cranes, and flat transporters. One of the most basic and fundamental resources is the work plate, where the block is made and assembled sequentially. The block spatial scheduling (BSS) is a dynamic assignment

that allocates blocks to suitable positions in the work plate at the right time. Meanwhile, all the constraints are satisfied. The selection of the processing position and time of any block would affect the subsequent block scheduling. The BSS problem as an NP-hard problem can be hardly found to be an accurate optimum solution. In practice, BSS is still designed empirically in many shipyards. An efficient automatic algorithm of BSS is imperative for productivity improvement.

Work plate is the most significant resource in the whole process of shipbuilding, and an efficient management system is critical for solving the block spatial scheduling problem. Therefore, scholars and designers have presented many algorithms. For instance, Lee et al. [1] proposed a block dynamic spatial layout algorithm based on a three-dimensional (3D) packing model and indicated that BSS should also consider the dynamic spatial layout of blocks, in addition to the traditional scheduling constraints like resource capacity and due date. The 3D packing model of BSS was constructed by adding a time axis to a two-dimensional (2D) allocation model developed by Lozano [2]. Park et al. [3] proposed

an efficient heuristic search procedure for spatial allocation by partial enumeration and decomposition. By extending the above algorithm [1], Koh et al. [4] decomposed the BSS solving process into an assembly sequencing phase and a spatial layout phase. Shin et al. [5] and Ryu et al. [6] conducted many works on the two phases to solve the BSS problem in Daewoo Shipbuilding Company. Shin et al. [5] defined a block allocation policy to determine the allocation sequence on the basis of a bottom-left-fill (BLF) heuristic method and presented a heuristic arrangement algorithm for automatic block arrangement. Ryu et al. [6] established a scheduling system based on an object-oriented methodology to support spatial planning and scheduling in block assembly shops and the workload balance of workshops. To obtain the optimal solution of the BSS problem, Zheng et al. [7] sequenced the blocks following the rule of earliest starting time + latest processing time (EST + LPT) and proposed a heuristic algorithm for dynamic spatial layout. Their experimental results show better performance. Zhuo et al. [8] proposed a hybrid model that contained 11 scheduling rules and the bottom-left- (BL-) based method of block allocation. Kwon and Lee [9] extended the two-phase approach to solve the BSS problem and developed a two-stage heuristic algorithm using dispatch priority rules and a diagonal fill spatial allocation method.

Spatial scheduling in shipbuilding is a complicated and comprehensive practical problem with many key points to be studied. To solve the spatial allocation problem, Lee [10] presented an efficient search method to locate rectangular and arbitrary-shaped objects within a 2D rectangular space. Koh et al. [11] proposed a Largest Contact Area (LCA) policy to select a suitable location for blocks and found that the block assembly scheduling system based on this policy was improved. Zhang and Chen [12] proposed an agglomeration algorithm (AA) which used 3D classification to cluster blocks linked closely in time and space into virtual blocks. Tao et al. [13] studied a combinatorial optimization problem of spatial scheduling and workforce assignment in shipbuilding. Park et al. [14] described HYPOS project and applied it successfully at a block paint shop in a shipbuilding company.

To date, many methods and algorithms have been proposed to solve the BSS problem from different perspectives, but the real efficiency of block spatial scheduling is still low. Thus, here we proposed a BCA, which has better use effect in dynamic block placement. However, since block spatial scheduling is an NP-hard problem, the scheduling of fixed building sequence rules does not play the role of optimization actually. Thus, GA was used to find out the optimal block building sequence under the condition of satisfying its time and space constraints.

The remainder of this paper is organized as follows. Section 2 describes the BSS problem and the key resource constraints of block processing and presents a mathematical model based on the above investigation. Section 3 introduces two optimization algorithms for BSS: best contact algorithm (BCA) and genetic algorithm (GA). To verify the performance of these algorithms, we establish a spatial scheduling system and conduct some experiments using real data from a shipyard. Section 4 shows the operational results compared

with traditional algorithms. Section 5 lists the summaries and conclusions.

2. Problem Description

2.1. Block Spatial Scheduling. With blocks as the shipbuilding units, each block is assigned with a separate processing technology and time attribute depending on the manufacturing requirements and delivery date. The given time attributes, including earliest starting time (EST), due date (DD), and processing time (PT), should be taken into account. EST is determined by the material delivery date and building priority, and DD can be derived through the hull erection schedule, while PT is a prevalue assigned by personnel experience and empirical data. Another important parameter is the latest starting time (LST), which can be calculated by DD minus PT. These parameters made by the design institute reflect the processing time and between-block relationship as procedure constraints. In addition, the geometric shape of blocks is regarded as rectangles since most of the blocks are rectangles and the remaining part can be expressed easily. During the assembly, no block overlaps physically with others or overhangs the boundary of the work plate.

The blocks are moved sequentially and continuously into the work plate for processing and assembly and then are moved out. The process is repeated until all blocks are processed. Moreover, the blocks, once they enter the field, cannot be moved from their location until the end of the process. And the placement of blocks can only be orthogonal to the work plate. An example of BSS is shown in Figure 1. Two blocks are separated by shape attributes called block 1 and block 2. They should be processed in a rectangular work plate which contains some preexisting blocks. Three scheduling schemes are compared as indicated. In plans 1 and 2, only one block can be allocated in the work plate in two days, while, in plan 3, because of the appropriate placement of block 2 on day 1, both blocks 1 and 2 can be arranged into the work plate on day 2. Obviously, plan 3 improves the space utilization significantly and shortens the makespan by optimizing assembly sequence and block placement. This is exactly the effect that the spatial scheduling aims to achieve.

A 3D packing model with time constraint is widely used to solve the spatial scheduling problem, since it can vividly describe the dynamic allocation of blocks. As shown in Figure 2, a 2D planar space is built to represent the physical dimensions of blocks and the work plate, while the z -axis is introduced to reflect time. In particular, the starting time of a block is its height at the z -axis and is constrained between EST and LST. The PT of a block is its own height in the space. In this model, the BSS problem is treated so as to determine the position (x, y, z) of each block, aiming to minimize the overall placement height.

2.2. Spatial Scheduling Model. The description above and Figure 2 show that, to solve the BSS problem, we have to find out the best (x, y, z) for all blocks. Specifically, (x, y) is the processing position of a block in the work plate, while z is its starting time that satisfies the time requirement. Since z should be limited within EST and LST, we describe

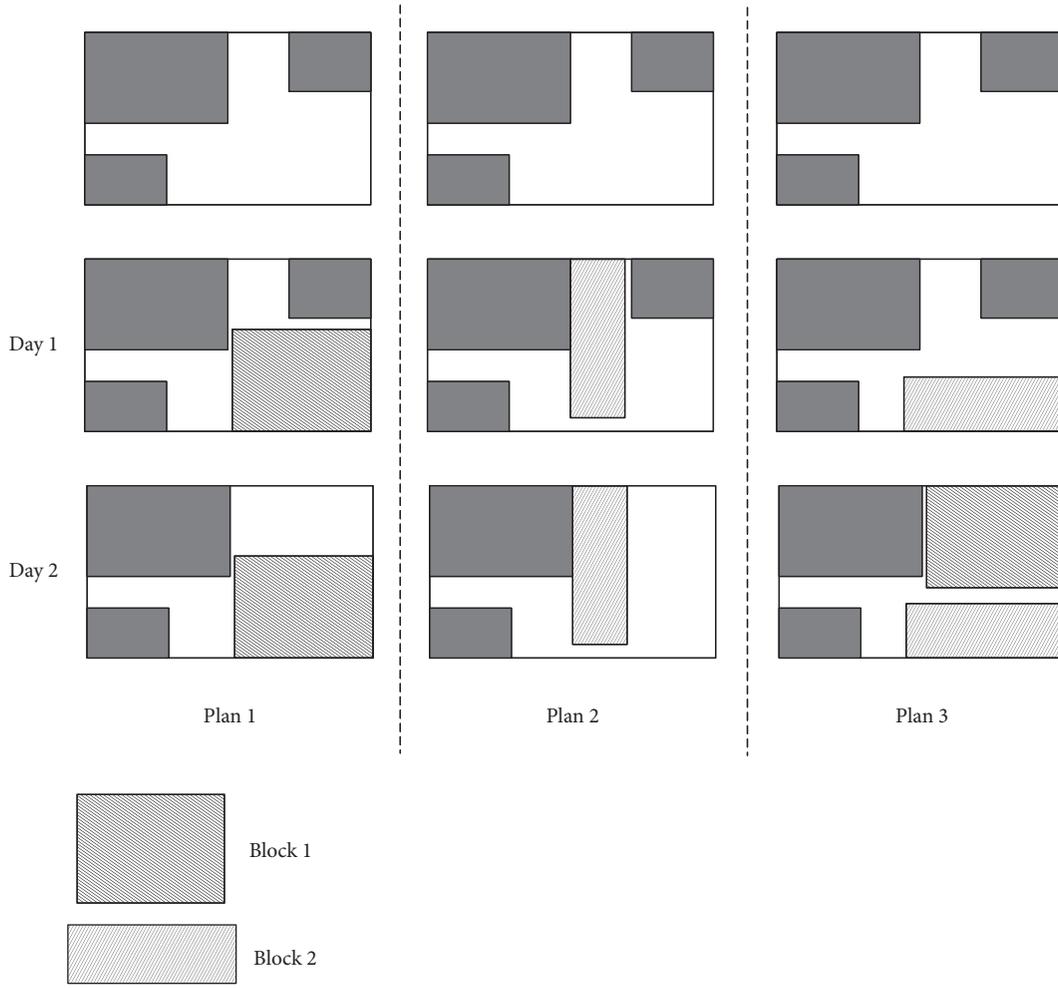


FIGURE 1: A comparison of different spatial schedules.

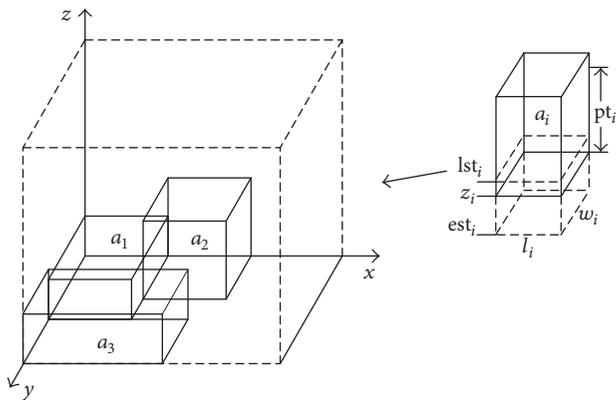


FIGURE 2: Three-dimensional packing model with time constraint.

z by introducing a variable DT . The DT is defined as the delayed time of a block after reaching the EST , or, namely, the difference from the starting time to the EST . The concrete mathematical model is built as follows:

$$\text{Objective: min } \{ \max (z_i + pt_i) \} \quad (1)$$

$$\begin{aligned} \text{Subject to: } & x_{a_i \in V_{lq}} + l_i \times (1 - r_i) + w_i \times r_i \\ & \leq L \times (1 - lt_{ij}) + x_{a_j \in V_i} \times lt_{ij} \end{aligned} \quad (2)$$

$$\begin{aligned} & y_{a_i \in V_{lq}} + w_i \times (1 - r_i) + l_i \times r_i \\ & \leq W \times (1 - bk_{ij}) + y_{a_j \in V_i} \times bk_{ij} \end{aligned} \quad (3)$$

$$dtr_i = dd_i - est_i - pt_i \quad (4)$$

$$z_i = est_i + dt_i \quad dt_i \in [0, dtr_i] \quad (4)$$

$$z_i + pt_i < z_j \quad \text{if } dd_i < est_j, \quad i \neq j. \quad (5)$$

Notation

Indices

i, j : block sequence number, $i, j = 1, 2, \dots, n$.

t : time.

V_{tq} : the set of blocks that are newly added to the work plate at time t .

V_t : the set of blocks that are processed within the work plate at time t and need to be processed continuously after time t .

Parameters

l_i : length of block a_i .

w_i : width of block a_i .

L : length of work plate.

W : width of work plate.

est_i : earliest starting time of block a_i .

dd_i : due date of block a_i .

pt_i : processing time of block a_i .

dtr_i : range of allowable delayed time of block a_i .

Variables

(x_{a_i}, y_{a_i}) : the bottom-left coordinates of a_i .

z_i : starting processing time of block a_i .

dt_i : delayed starting time of block a_i .

r_i : rotation of block a_i . If the longer side of a_i is parallel to the longer side of the work plate, $r_i = 0$. Otherwise, the block is rotated at an angle of 90° and $r_i = 1$:

$$lt_{ij} = \begin{cases} 1, & \text{If block } a_i \text{ is on the left of block } a_j \\ 0, & \text{Otherwise} \end{cases} \quad (6)$$

$$bk_{ij} = \begin{cases} 1, & \text{If block } a_i \text{ is on the back of block } a_j \\ 0, & \text{Otherwise.} \end{cases}$$

The optimization objective in BSS is to minimize the makespan as shown in (1). Equation (2) guarantees that the newly allocated block does not overlap with preexisting blocks or with the boundary while considering the rotation. Equation (3) defines the range of allowable delayed time. In (4), the starting time of each block is replaced by its earliest starting time and the delayed starting time. Specifically, due to the time attributes of each block, the selection of its delayed time could be limited to a range as described by (3). As the assembly relationship between blocks, the sequence of block processing is expressed in (5): if the earliest starting time of a_j is later than the due date of a_i , the starting processing time of block a_j must be later than the completion of block a_i .

In this mathematical model, the input and output of the spatial schedule are given as follows:

$$\begin{aligned} \text{Input} &= \{a_i, l_i, w_i, pt_i, est_i, dtr_i\} \\ \text{Output} &= \{a_i, x_i, y_i, r_i, z_i\}. \end{aligned} \quad (7)$$

3. Spatial Scheduling Algorithm and System

The block spatial scheduling is a dynamic process, so the selection of processing position and starting time of any block would affect the whole scheduling. Just like a 3D packing problem, the optimal block placement way at any time point would not necessarily mean the final overall placement effect is the optimal. Thus, the optimal solution of the BSS problem could be found only from the global perspective. Starting from the two phases of spatial block allocation and assembly sequence, we propose a 2D spatial allocation method, BCA, that is suitable for dynamic scheduling and thereby use GA to optimize the assembly sequence.

3.1. Spatial Allocation. The spatial allocation aimed to place the blocks into a work plate which already had blocks, so as to improve the space utilization rate. Heuristic allocation methods have already been studied. Baker et al. [15] proposed the bottom-left (BL) allocation method which places bins sequentially in a bottom-leftmost position. Chazele [16] developed the bottom-left-fill (BLF) method based on BL method. This method can find a location for each bin by searching the feasible lowest bottom-left placement point. Hopper and Turton [17] demonstrated that the BLF method is better than BL method and is widely used in packing problem along with various algorithms. Kwon and Lee [9] proposed a diagonal fill method pointing at dynamic placement of blocks in the work plate. It considers both the bottom-left and top-right points of placed blocks instead of bottom-left points only. A comparison of these three placement methods is shown in Figure 3.

Moreover, research on spatial allocation as a pure 2D bin packing problem has brought about some achievements. Huang et al. [18] proposed a deterministic heuristic algorithm that evaluates all available allocation places and selects the best scheme for a newly allocated bin. By extending this idea, He et al. [19, 20] developed BFA and LIF algorithms to solve different 2D packing problems and achieved good results. Thus, by using the idea of evaluating and selecting the possible block placement positions, we proposed BCA that was more suitable for BSS problems.

When a block is to be placed into a working platform, BCA computes the contact degree of all suitable positions in the space and selects the position with the best contact degree for block placement. The contact degree is defined as the position relationship between the newly allocated and the preexisting blocks (the work plate was considered as a special existing block), as shown in (8). In this evaluation degree, n_{ik} and p_{ik} imply the contact compactness between blocks, of which a larger value means higher d_{ik} . x and y are defined as the smaller the better. In the comparison and choice of d_{ik} , the priority of parameters is $n_{ik} > p_{ik} > x_{ik} > y_{ik}$.

$$d_{ik} = (n_{ik}, p_{ik}, x_{ik}, y_{ik}) \quad k \in W_t. \quad (8)$$

Notation

k : an available placement position of a_i .

W_t : set of all available placement positions of a_i at time t .



FIGURE 3: Comparison of different allocation methods after the addition of a bin.

d_{ik} : contact degree of a_i at position k .

n_{ik} : number of complete coincident edges of block a_i with preexisting blocks at position k , $n_{ik} \in \{1, 2, 3, 4\}$.

p_{ik} : length of coincident edge of block a_i with preexisting blocks at position k .

(x_{ik}, y_{ik}) : center coordinate of block a_i at the position k .

To describe the selection process for block placement, an example is given in Figure 4. There are two preexisting blocks a_j, a_{j+1} in the work plate and three succeeding placement positions when a new block a_i is added. In Figure 4(b), only one edge of block a_i completely coincides with other blocks, while there are two edges in both Figures 4(c) and 4(d), indicating that the allocations in Figures 4(c) and 4(d) are better. Compared with the central coordinate x of block a_i in Figure 4(c), it is smaller in Figure 4(d). So, the placement among the lots in Figure 4(d) is the best. Consequently, the best placement of each block can be selected by the calculation of contact degree.

Shipbuilding is a dynamic process in which the blocks are placed and removed continuously in the work plate. BCA considers the position relationship of blocks in order to maximally reduce the scattered spaces and improve the compactness of blocks. A comparison with traditional methods is shown in Figure 5. BCA deterministically evaluates every placement position of the newly allocated block and thus obtains an optimal result without extra strategies.

3.2. Assembly Sequencing. Different priority rules such as EST and EST + LPT and the urgency of blocks have been used to study block assembly sequencing. However, BSS is a dynamic NP-hard problem, so the use of a fixed sequencing rule will inevitably reduce the scheduling efficiency. Thus, we selected the starting time as a variable and used GA to globally operate and optimize the assembly sequence. Thereby, the spatial scheduling system is established.

3.2.1. Encoding Scheme. Assembly sequence can be figured out by determining the starting process time (z_i) of each block. As shown in (4), z_i is composed of the earliest starting time (est_i), which is a constant with the block's attributes, and delayed starting time (dt_i), which ranges from 0 to dtr_i . Therefore, we treated dt_i with GA, so as to find out the optimal assembly sequence. A chromosome with the time information of blocks is obtained as follows:

$$C = \{dt_1, est_1, dtr_1; dt_2, est_2, dtr_2; \dots; dt_i, est_i, dtr_i; \dots\} \quad (9)$$

Subsequently, the corresponding assembly sequence is calculated as follows:

$$S = \{\dots, a_{z_i}, \dots, a_{z_j}, \dots\} \quad z_i < z_j. \quad (10)$$

Also, the blocks are arranged from large to small sizes if the starting time points of the blocks are the same in the sequencing process.

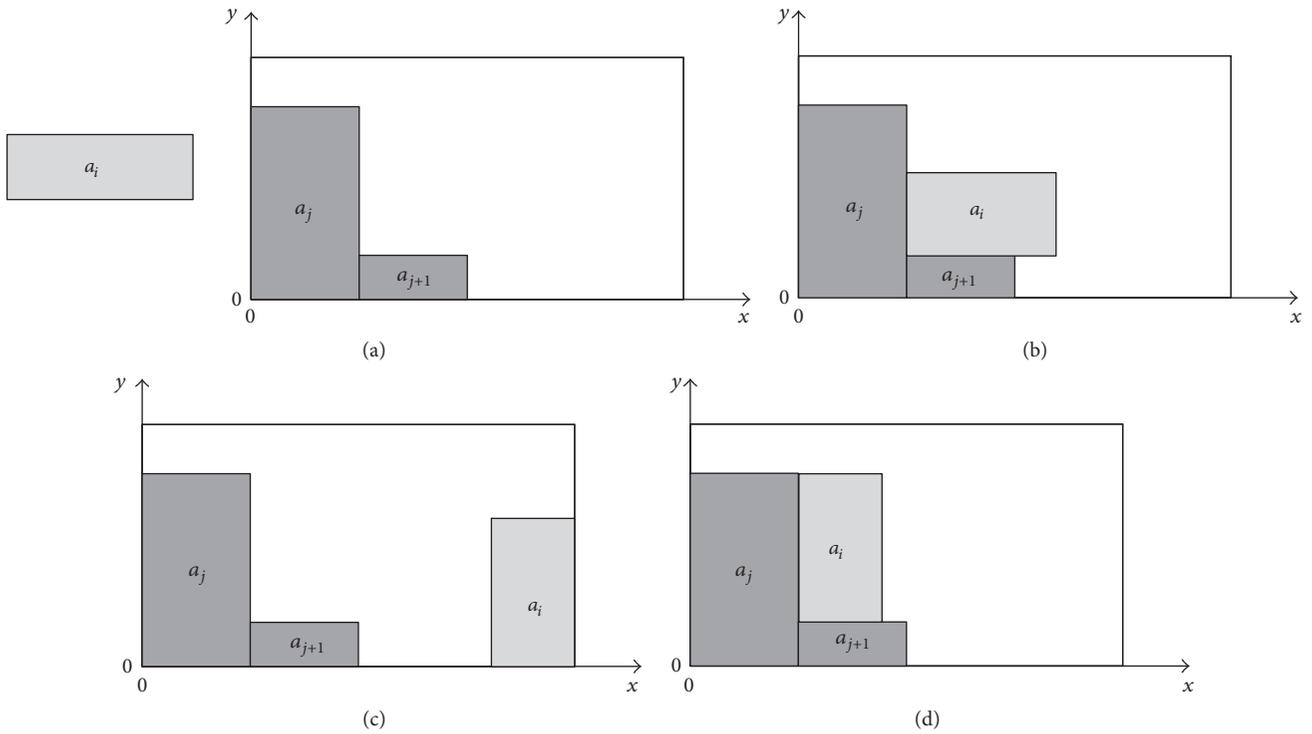


FIGURE 4: An example of placement selection using the BCA.

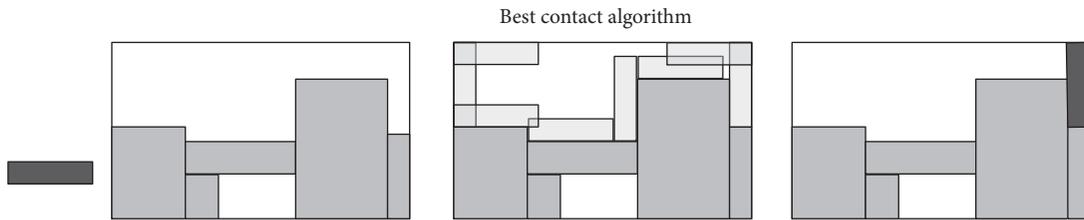


FIGURE 5: Spatial allocation in BCA after the addition of a bin.

3.2.2. Initial Solution. To quickly get a good solution, especially to the problem needing huge computation, we often set the initial solution in GA as the relatively optimal one. Hence, the initial solution is established by the EST rule (the assembly sequence of blocks is built according to the earliest starting time from early to late) in this algorithm, which is widely used in assembly sequencing. The process is given as follows:

$$S_{\text{ini}} = \{\dots, a_{\text{est}_i}, \dots, a_{\text{est}_j}, \dots\} \quad \text{est}_i < \text{est}_j. \quad (11)$$

The initial solution S_{ini} means an assembly sequence is constructed such that the EST of a block is its starting time. Its expression in chromosomes is $dt_i = 0$. Thus, the initial population is designed as follows:

$$C_{\text{ini}} = \{0, \text{est}_1, \text{dtr}_1; 0, \text{est}_2, \text{dtr}_2; \dots; 0, \text{est}_i, \text{dtr}_i; \dots\}. \quad (12)$$

3.2.3. Offspring Generation. This algorithm adopts iterative computation from dt_i of each population and applies the

crossover and mutation operators to produce offspring. These operations are illustrated in Figure 6.

The crossover operator, which applies to pairs of chromosomes as shown in Figure 6(a), stochastically selects two parents and exchanges a part of genes in the same position. The mutation operator, which applies to single individuals as shown in Figure 6(b), selects a gene dt_4 by probability and replaces it by any integer from dtr_4 . Here, multipoint crossover and mutation are used in offspring generation owing to the relatively long gene sequences of the blocks.

3.2.4. Selection. During iteration, the makespan of an assembly sequence is considered as the fitness corresponding to the chromosome. It can resolve the dynamic block spatial allocation. Here, a selection method is presented to obtain the chromosome with the best fitness. Firstly, the middle offspring is generated from the parent by using the crossover and mutation operators. Then, a new population is constructed from the parent and the middle offspring as follows:

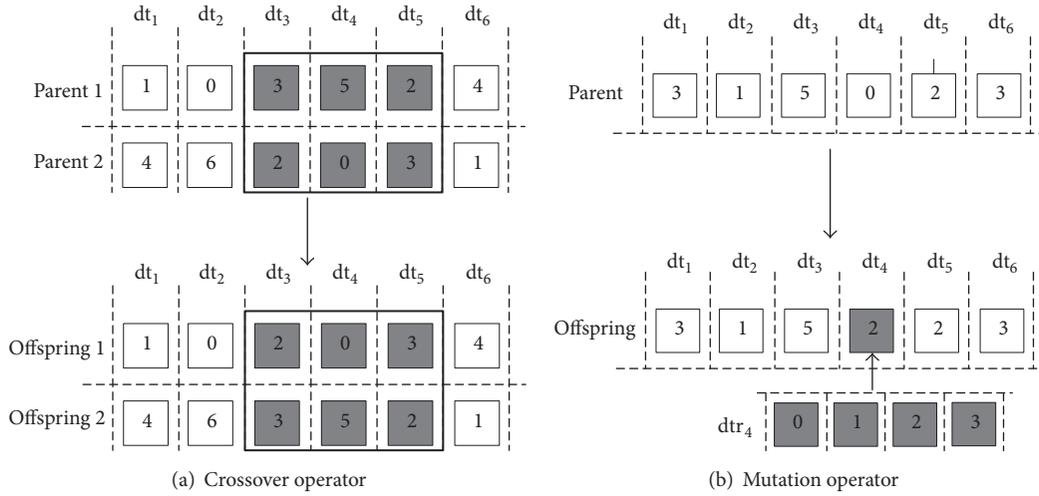


FIGURE 6: Example of crossover and mutation operators.

TABLE 1: Data structure of scheduling block.

n	Mean ($pstr_i$)	Std. ($pstr_i$)	Mean (pt_i)	Std. (pt_i)	Mean (l_i)	Std. (l_i)	Mean (w_i)	Std. (w_i)
157	7.02	3.66	25.83	4.55	14.06	4.38	9.7	3.06

TABLE 2: Performance comparison between BLF and BCA.

Algorithm	P_{u_max}	P_{u_min}	P_{u_ave}	R_{t_max}	R_{t_min}	R_{t_ave}
BLF	90.8%	80.6%	86.4%	1.68	1.88	1.75
BCA	93.7%	83%	89.5%	4.59	3.82	4.26

- (1) Select the chromosomes in the middle offspring whose fitness is better than the parent's and accept them as a part of the new population.
- (2) Build a completely new population from the remaining middle offspring and the parent with different probabilities.
- (3) Replace the worst solutions in the new population with the best solutions in the parent.

GA is actually a searching method based on iteration and selection. Figure 7 shows a spatial scheduling system established by the dynamic spatial allocation algorithm and GA. The chromosome and fitness in this system represent the assembly sequence and makespan, respectively, which are constantly generated and selected. When an assembly sequence with the best fitness is solved, a related schedule of block processing can be outputted. Noticeably, the block assembly sequence modestly reflects the building relationship among blocks. Thus, the processing sequence, once determined, should not be changed during the placement. In other words, when a candidate block cannot be placed into the work plate, we run “day++” until the work plate could accommodate this block.

4. Experiment and Implementation

To benchmark the performances of BCA and the established system, we performed and compared experiments with representative BLF and EST that are widely used in the existing spatial scheduling systems. All computations and experiments were carried out on a workstation with an Inter(R) Xeon(R) CPU E5-2650 v3 and 64 GB memory using Matlab 2015a. The performance indicators were determined by averaging the data in ten tests.

Test data were collected from a large shipbuilding company in Shanghai, China, including 157 continuously processed real blocks and a work plate of $24 \times 138 \text{ m}^2$. The major data structure of the block comprises the delayed starting time range (dtr_i), processing time (pt_i), and shape attributes, expressed mathematically in Table 1.

4.1. Performance of BCA. Ten groups of blocks were allocated in the initial (empty) work plate by BLF and BCA separately. Each group was composed of 30 blocks that were randomly selected from a total of 157 blocks. The results of spatial utilization ratio (P_u) and operation time (R_t) are shown in Table 2. Clearly, BCA outperforms BLF, although the computation time is prolonged as more locations are searched. A pair of allocation results in the same group are obtained as shown in Figure 8.

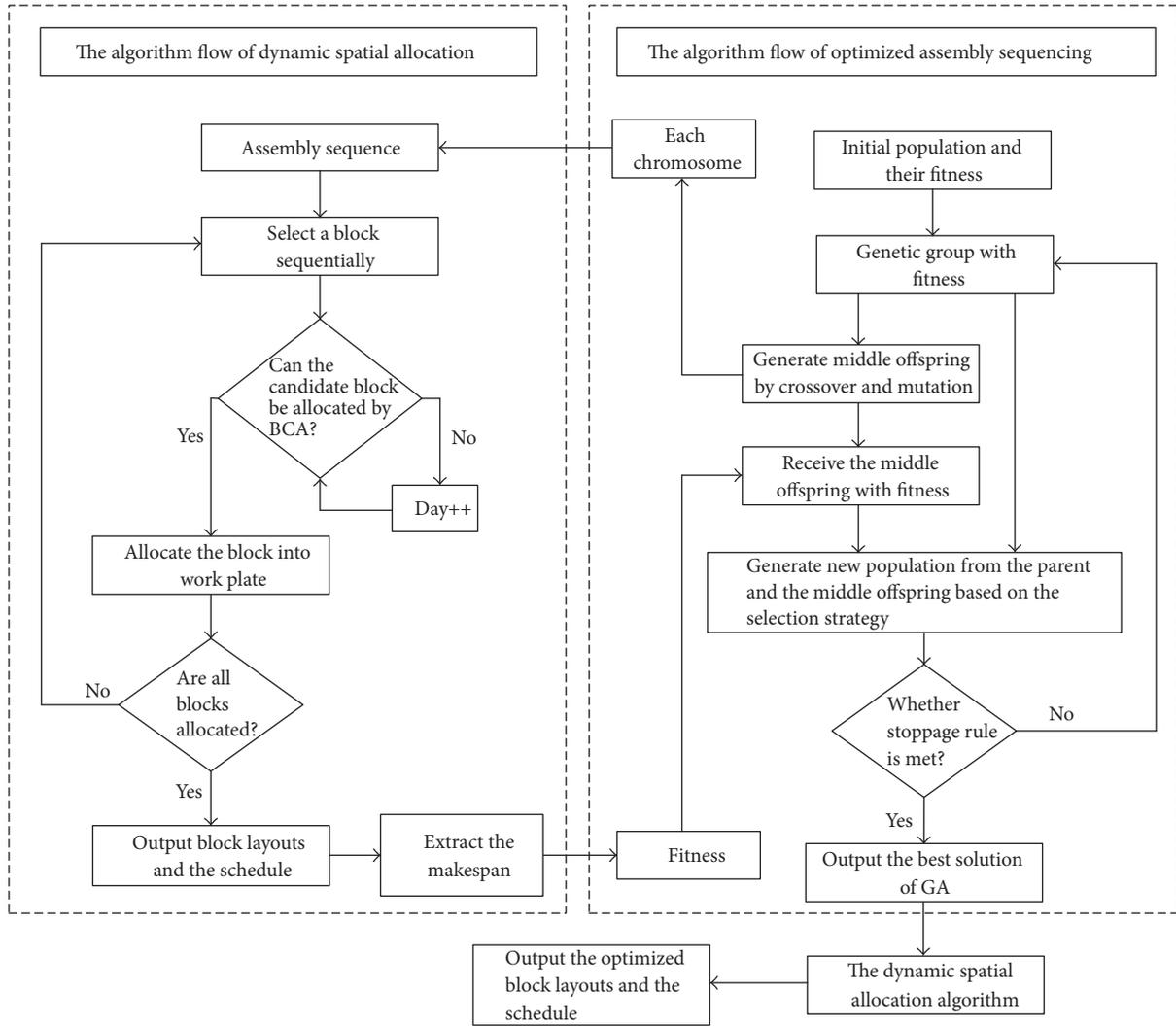


FIGURE 7: Spatial scheduling flowchart.

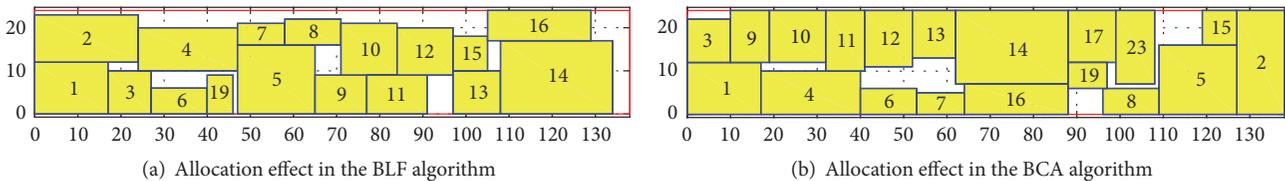


FIGURE 8: Comparison of allocation effect between BLF and BCA.

The order and location of each block are shown in Figure 8. BCA achieves the optimization of 2D block allocation by selecting placement with the best evaluation for each block. When the blocks are placed into and removed out of the work plate, this algorithm reduces the scattered areas in the space by using global search and makes the contact between blocks closer and more regular. Such improvement helps to improve the space utilization rate and promotes the formation of a larger residual space for the subsequent block placement.

4.2. Performance of the Established Spatial Scheduling System. In actual shipbuilding, scheduling is a process to keep operation continuous. To examine the performance of the proposed scheduling system, we designed a virtual experiment: lay out the real-world blocks into the work plate one by one until all blocks are processed, while the nonspatial resources are adequate. To reflect the real scheduling of blocks, we introduced two definitions: the start time of rolling plan (ST_r), which is the day of the first block being allocated to the work plate after the initial allocation, and the finish

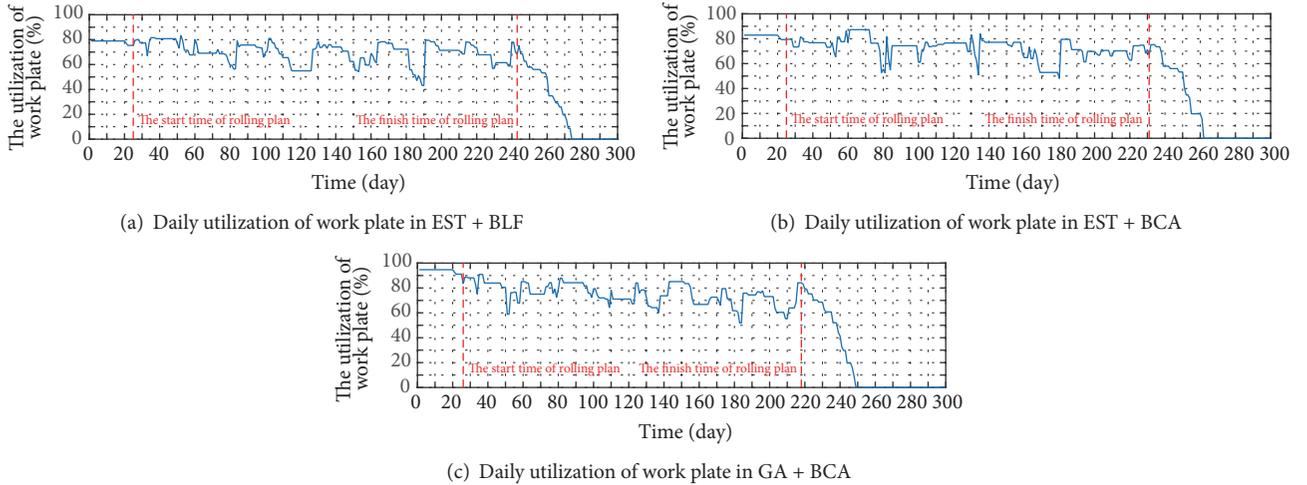


FIGURE 9: Daily utilization of work plate.

TABLE 3: Performance results' comparison.

Algorithm	ST_r (day)	FT_r (day)	M (day)	S_i (%)	S_u (%)	R_t (s)
EST + BLF	25	243	273	78.93	69.9	17
EST + BCA	25	231	261	82.91	72.8	42
GA + BCA	26	218	248	94.68	75.4	19764

time of rolling plan (FT_r), which is the day of the last block being allocated to the work plate. During the period, the blocks are processed continuously, like the normal rolling plan of the shipyard. Thus, the makespan and spatiotemporal utilization of the work plate are considered and calculated in this period. In comparison of different spatial scheduling algorithms, the performance indicators are listed in Table 3, including makespan (M), utilization ratio of initial work plate (S_i), average spatiotemporal utilization ratio (S_u), and runtime (R_t). In particular, S_u which is the most direct and accurate data in the experiment is used as the most important influencing factor to reflect the makespan.

GA is set to perform with a population number of 40 and 30 iterations. The scheduling results through EST + BLF, EST + BCA, and GA + BCA are shown in Table 3. Compared with EST + BLF, GA + BCA leads to 9.15% reduction of M and 7.87% increment of S_u . The corresponding scheduling system outperforms others whose schedules are produced according to the conventional BLF placement method and the fixed-rule EST sequencing method. Actually, the average spatiotemporal utilization ratio of 75.4% is higher than the results and actual usage effects of other algorithms. Although the computation time of the system is prolonged by the calculation of GA, it is acceptable for the generation of a small-scale or medium-scale production plan in continuous scheduling.

Figure 9 shows the daily utilization of work plate in three algorithms. The utilization rate curve clearly shows that large fluctuation exists in the scheduling and is a major influencing factor on the whole makespan. The cause for this fluctuation is the extra waiting time when a block to be arranged into

the work plate cannot be placed due to the inadequate placement. This extra waiting time is expressed as day++ in the algorithm. Regarding this key problem, we proposed BCA that could better lay out the blocks dynamically. And GA was used to globally optimize the building sequence. As illustrated in Figure 9(c), the number of days is reduced with utilization ratio less than 60% and is increased with utilization ratio more than 80%. Results show that BCA + GA can effectively improve the space utilization rate and reduce the extra waiting time during block scheduling. As a result, the overall block completion time is shortened.

5. Conclusions

Block spatial scheduling takes place in almost every stage of shipbuilding, which plays the key role in the successful operations of a shipyard. However, the scheduling level in the work plate is still very low. As is well known, it is a bottleneck against the improvement of shipbuilding efficiency. Here, according to the time and space attributes in block assembly, a dynamic 3D mathematical model was established. To resolve and optimize the BSS problem, we proposed an efficient allocation algorithm, BCA, and used GA to determine a good block assembly sequence. After that, computation and experiments were carried out by the data of 157 blocks from an actual shipyard. BCA improves space utilization and reduces scattered space in the work plate, which are more appropriate for dynamic block layout. The BCA + GA scheduling system could provide the suitable building position and starting time of each block. Compared with the conventional EST + BLF algorithm, the proposed algorithm not only generated a block

assembly scheduling plan according to certain rules, but also achieved optimization from a global perspective.

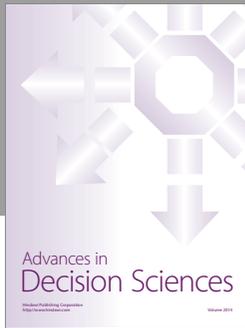
In summary, the BCA + GA algorithm is capable of optimizing block spatial scheduling and improves the shipbuilding efficiency. It may also be extended to other similar manufacturing industries such as airplanes and special vehicle types. Future research should focus on more resource-constrained and more accurate scheduling models.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] K. J. Lee, J. K. Lee, and S. Y. Choi, "A spatial scheduling system and its application to shipbuilding: DAS-CURVE," *Expert Systems with Applications*, vol. 10, no. 3-4, pp. 311–324, 1996.
- [2] P. T. Lozano, "Spatial planning: a configuration space approach," *IEEE Transactions on computers*, vol. 32, no. 2, pp. 108–120, 1983.
- [3] K. Park, K. Lee, S. Park, and S. Kim, "Modeling and solving the spatial block scheduling problem in a shipbuilding company," *Computers and Industrial Engineering*, vol. 30, no. 3, pp. 357–364, 1996.
- [4] S.-G. Koh, J.-C. Park, Y.-S. Choi, and C.-M. Joo, "Development of a block assembly scheduling system for shipbuilding company," *IE Interfaces*, vol. 12, no. 4, pp. 586–594, 1999.
- [5] J. G. Shin, O. H. Kwon, and C. Ryu, "Heuristic and metaheuristic spatial planning of assembly blocks with process schedules in an assembly shop using differential evolution," *Production Planning and Control*, vol. 19, no. 6, pp. 605–615, 2008.
- [6] C. Ryu, J. G. Shin, O. H. Kwon, and J. M. Lee, "Development of integrated and interactive spatial planning system of assembly blocks in shipbuilding," *International Journal of Computer Integrated Manufacturing*, vol. 21, no. 8, pp. 911–922, 2008.
- [7] J. Zheng, Z. Jiang, Q. Chen, and Q. Liu, "Spatial scheduling algorithm minimising makespan at block assembly shop in shipbuilding," *International Journal of Production Research*, vol. 49, no. 8, pp. 2351–2371, 2011.
- [8] L. Zhuo, D. C. K. Huat, and K. H. Wee, "Scheduling dynamic block assembly in shipbuilding through hybrid simulation and spatial optimisation," *International Journal of Production Research*, vol. 50, no. 20, pp. 5986–6004, 2012.
- [9] B. Kwon and G. M. Lee, "Spatial scheduling for large assembly blocks in shipbuilding," *Computers and Industrial Engineering*, vol. 89, pp. 203–212, 2015.
- [10] K. J. Lee, "Sufficient search space for spatial expert systems," *Expert Systems with Applications*, vol. 19, no. 1, pp. 1–8, 2000.
- [11] S. Koh, C. Eom, J. Jang, and Y. Choi, "An improved spatial scheduling algorithm for block assembly shop in shipbuilding company," in *Proceedings of the 3rd International Conference on Innovative Computing Information and Control (ICICIC '08)*, Dalian, China, June 2008.
- [12] Z. Zhang and J. Chen, "Solving the spatial scheduling problem: a two-stage approach," *International Journal of Production Research*, vol. 50, no. 10, pp. 2732–2743, 2012.
- [13] N.-R. Tao, Z.-H. Jiang, and L. Zhen, "Spatial scheduling and workforce assignment problem in block assembly shop of shipbuilding," *Asia-Pacific Journal of Operational Research*, vol. 31, no. 1, Article ID 1450006, 19 pages, 2014.
- [14] C. Park, K.-H. Chung, J.-C. Park, K.-K. Cho, T.-H. Baek, and E.-I. Son, "A spatial scheduling application at the block paint shop in shipbuilding: the HYPOS project," *Production Planning & Control*, vol. 13, no. 4, pp. 342–354, 2002.
- [15] B. S. Baker, J. Coffman, and R. L. Rivest, "Orthogonal packings in two dimensions," *SIAM Journal on Computing*, vol. 9, no. 4, pp. 846–855, 1980.
- [16] B. Chazele, "The bottom-left bin-packing heuristic: an efficient implementation," *IEEE Transactions on Computers C*, vol. 32, no. 8, pp. 697–707, 1983.
- [17] E. Hopper and B. C. H. Turton, "Empirical investigation of metaheuristic and heuristic algorithms for a 2D packing problem," *European Journal of Operational Research*, vol. 128, no. 1, pp. 34–57, 2001.
- [18] W. Huang, D. Chen, and R. Xu, "A new heuristic algorithm for rectangle packing," *Computers & Operations Research*, vol. 34, no. 11, pp. 3270–3280, 2007.
- [19] K. He, W. Huang, and Y. Jin, "An efficient deterministic heuristic for two-dimensional rectangular packing," *Computers & Operations Research*, vol. 39, no. 7, pp. 1355–1363, 2012.
- [20] K. He, P. Ji, and C. Li, "Dynamic reduction heuristics for the rectangle packing area minimization problem," *European Journal of Operational Research*, vol. 241, no. 3, pp. 674–685, 2015.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

