

Research Article

A 3D Offline Packing Algorithm considering Cargo Orientation and Stability

Xianpei Wang , Li Gong , Haocheng Zhao, Bowen Li, and Meng Tian 

Electronic Information School, Wuhan University, Wuhan, 430072 Hubei Province, China

Correspondence should be addressed to Xianpei Wang; xpwang@whu.edu.cn

Received 10 November 2022; Revised 2 January 2023; Accepted 16 January 2023; Published 29 March 2023

Academic Editor: Kavita Pandey

Copyright © 2023 Xianpei Wang 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.

The box packing problem can be generalized as placing a batch of cargos with a specified number of different physical characteristics into a specified box. Suppose that a batch of cuboid cargos of different sizes are to be loaded into a batch of boxes of the same type, the cargos have constraints such as orientation and stability. Taking the mean value of the reciprocal of space utilization as the objective function, this paper designs a hybrid genetic algorithm that combines genetic algorithm and tabu search algorithm. Aiming at the information of the packing sequence and the rotating state of the box in the packing scheme, a two-stage real number encoding method and decoding method based on random keys are designed, and a crossover operation based on partial random keys and uniform crossover is designed. In order to convert the solution searched by the optimization algorithm into the actual packing scheme, a heuristic loading algorithm is designed while using the positioning rule of the lower left corner, the space selection rule of the minimum space, and the division and merging rules of the remaining space. In the early stage, the roulette method was used to strengthen the global search ability, and in the later stage, the optimal preservation strategy was used to speed up the algorithm convergence. To make up for the shortcomings of the genetic algorithm's weak local search ability and slow convergence speed, the tabu search algorithm was used as a mutation operation in the genetic algorithm. The solution in the generation is used as the initial solution of the tabu search algorithm, and the search process is carried out. Finally, this paper tests the proposed hybrid algorithm on 6 groups of weakly heterogeneous and strongly heterogeneous data in the BR dataset. The results prove that the proposed algorithm can reduce the usage of boxes.

1. Introduction

The bin packing problem has a long history and has been extensively studied by scholars Fazili et al. [1], such as the one-dimensional bin packing problems represented by resource scheduling, time scheduling, and load management, the two-dimensional bin packing problem represented by fabric cutting and sheet processing, and the three-dimensional bin packing problem represented by container loading and vehicle loading. Among them, three-dimensional bin packing problem is one of the most

concerned combinatorial optimization problems Sridharan and Dominic [2]. The main goal is to find a space allocation method that can achieve the highest space utilization or the lowest cost [3], which can be divided into container packing problem, box packing problem, and back packing problem (Kyungdaw et al. [4]; Mhand et al. [5]).

At present, the algorithms for solving the bin packing problem include precise algorithms, heuristic algorithms, intelligent optimization algorithms, and other algorithms. The exact algorithm uses the upper and lower limits of each parameter to guide the search in feasible solutions and find

the optimal solution. Commonly used precise algorithms are as follows [6, 7]: branch pricing algorithm [8], branch solution price cutting algorithm [9], and backtracking method [10]. For the classic combinatorial optimization problem, the precise algorithm can accurately obtain the global optimal solution in the bin packing problem; it is only suitable for small packing cases. When the complexity of the problem increases exponentially as the number of cargos in the problem becomes larger, the traversal search process cannot be implemented in a limited time.

Therefore, heuristic algorithm has become one of the important ways to solve this series of problems. The heuristic algorithm is a method constructed based on human experience, which includes the order of cargos packing and the rules of space allocation in the box [11]. Common methods include layered method, wall method, stacked tower method, composite block method, stacking method, and hole degree method [12, 13]. The number of iterations of the heuristic algorithm is proportional to the number of construction elements of the problem solution but has nothing to do with the size of the solution space, so its calculation speed is usually very fast. However, the heuristic information introduced by the general heuristic algorithm depends on personal experience and also cannot effectively avoid the combinatorial explosion problem. More artificial intelligence algorithm has achieved great results in many fields, including solving combinatorial optimization problems [14, 15]. The intelligent optimization algorithm is an algorithm inspired by a certain phenomenon in nature, which mainly includes genetic algorithm [16], particle swarm algorithm [17], simulated annealing algorithm [18], and ant colony algorithm [19]. The artificial intelligence algorithm can cope with large bin packing problems and find better optimal solutions with low requirements for the expertise of the algorithm designer, but the convergence speed of the intelligent optimization algorithm is greatly affected by the initial solution and encoding method.

In addition to the above three types of algorithms, deep learning and reinforcement learning have been applied in the bin packing problem. Wang et al. proposed a cooperative memetic algorithm [20], and Pan et al. proposed a knowledge-based two-population optimization algorithm [21] to minimize total energy consumption and total tardiness simultaneously. Zhao et al. proposed the mixed-integer linear programming model to produce an effective constructive heuristic [22] and a hyperheuristic with Q-learning to address the energy-efficient distributed blocking flow shop scheduling problem [23]. These methods have better performance than traditional algorithms, but the use of these methods requires a large data set to train the model, and the computational speed is slower than traditional algorithms and requires more computer hardware.

Suppose to pack the cuboid cargos of different sizes, direction constraints, and stability constraints into boxes with the same type and unlimited quantity, the paper establishes the objective function with the least number of boxes and designs a hybrid genetic algorithm (HGA) that combines genetic algorithm, tabu search algorithm, and heuristic algorithm. The paper focuses on the 3D offline bin packing problem that requires multiple boxes to complete loading

in practical situations. Our main research work includes as follows:

- (1) We model the bin packing problem by identifying six rotational states of the cargos and constraints
- (2) We design a two-segment real number encoding method and decoding method based on a random key to determine the packing order of the cargos
- (3) We design the HGA to improve the space utilization of boxes
- (4) We design comparison experiments on the BR data to verify the rationality of the proposed algorithm

The remainder of the paper is structured as follows. In Section 2, we simplify the three-dimensional offline bin packing problem and establish the mathematical model through the rotational states of the cargos and the constraints. In Section 3, we divide the HGA into three parts for detailed analysis. In Section 4, we present and discuss experimental results. In Section 5, we conclude our work.

2. Problem Description

There are n cuboid cargos of different sizes, which are packed into a set of boxes with a fixed size of length L , width W , and height H . Some cargos have orientation requirements.

2.1. Related Assumptions

- (1) All cargos are cuboids, and the density of each part is equal. In order to facilitate modeling and analysis, the actual weight distribution is assumed to be uniform weight distribution
- (2) There are no cargos that cannot be load bearing, or cargos that need to be loaded separately, or cargos that cannot be squeezed
- (3) The size of any cargo is smaller than the size of the box, and the weight of any cargo is within the load range of the box
- (4) The cargos are in the box, and the direction of placement must be consistent with the direction of the box
- (5) The priority of the cargo is the same, there is no cargo that needs to be loaded or unloaded first, and the loading order of the cargo is only related to the calculated packing plan

2.2. Constraints

- (1) *Volume Constraint.* The box has a shell with a fixed size
- (2) *Guillotine Constraint.* If the cargos are not allowed to be separated, the cargos are regarded as an inseparable whole. Any one item can only be placed in one

box. As shown in equation (1), for the cargos with serial number i , in N boxes, only one y_{ij} is 1

$$\sum_{j=1}^N Y_{ij} = 1, \quad (1)$$

where y_{ij} indicates whether the cargos with the serial number are placed in the box with the serial number j ; if the cargos with the serial number i are placed in the box with the serial number j , $y_{ij} = 1$; otherwise, $y_{ij} = 0$

- (3) *Load Constraint.* Each box has a load limit, and the total weight of the cargos in the box cannot exceed the upper limit. The sum of the weights of the cargos loaded in each box is less than the upper limit of the box, as shown in the following equation:

$$\sum_{k=1}^{\text{num}_j} m_k \leq M, \quad (2)$$

where M is the upper limit of the load of the box, which represents the weight of the j th cargo in the cargo that needs to be loaded in the k th box in the loading plan

- (4) *Rotation Constraint.* Some cargos have restrictions on the way they are placed, so when designing a packing plan, the rotation constraints of the cargos should be met. For example, there is an upward sign on the package of glassware; that is, during the loading process, the sign can only be vertically upward but can be rotated in the horizontal direction
- (5) *Center of Gravity Constraint.* After packing the cargos, the center of gravity of the box should be within a certain range
- (6) *Stability Constraint.* During the loading process, the cargos must be placed on the bottom of the box or on top of other cargos. According to the size of the supported area on the bottom surface of the cargos, the stability can be divided into partial support and complete support. Among them, partial support means that at least part of the bottom of the cargo needs to be supported (as shown in Figure 1), and full support means that the bottom of the cargo needs to be fully supported (as shown in Figure 2)
- (7) *Placement Constraint.* The placement of the cargos in the box can only be orthogonal to the box; the six sides of the cargos are parallel to or overlap with the six sides of the box. The situation where the cargos and the box are crossed is not allowed; each side of the cargos should be parallel to one of the sides of the box as shown in Figure 3)

To maximize the loading capacity of the box and reduce the number of boxes used, the cargos can be placed in the

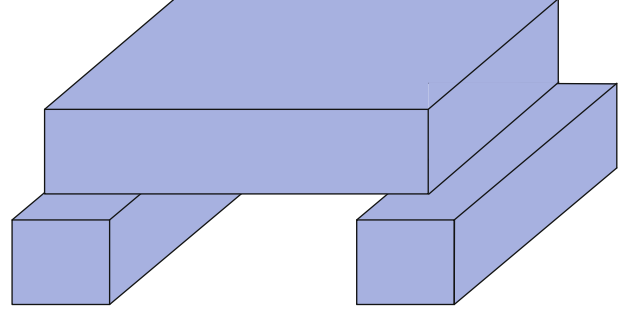


FIGURE 1: Schematic diagram of partial support.

box only in an orthogonal way to the box, where each side of the cargos should be parallel to one of the sides of the box.

$$\begin{cases} l_{xi} + l_{yi} + l_{zi} = 1, \\ w_{xi} + w_{yi} + w_{zi} = 1, \\ h_{xi} + h_{yi} + h_{zi} = 1, \\ l_{xi} + w_{xi} + h_{xi} = 1, \\ l_{yi} + w_{yi} + h_{yi} = 1, \\ l_{zi} + w_{zi} + h_{zi} = 1. \end{cases} \quad (3)$$

In the equation, l_{xi} indicates whether the length of the i th cargo is parallel to the x -axis of the box; if it is parallel, the value is 1; otherwise, it is 0. l_{yi} , l_{zi} , w_{xi} , w_{yi} , w_{zi} , h_{xi} , h_{yi} , and h_{zi} are similar to l_{xi} .

For the cuboid cargo and the cuboid box, there are six relative rotation states between the two. If the length, width, and height of the cargos are l , w , and h , respectively, and the length, width, and height of the box are x , y , and z , then there are six states of relative rotation between the two, i.e., $(x-l, y-w, z-h)$, $(x-l, y-h, z-w)$, $(x-w, y-l, z-h)$, $(x-w, y-h, z-l)$, $(x-h, y-l, z-w)$, and $(x-h, y-w, z-l)$. The volume constraint that the cargo in the rotating state should satisfy is shown in equation (4). The rotation constraints for the other five rotation states are similar to

$$\begin{cases} x_i + l_i \leq L, \\ y_i + w_i \leq W, \\ z_i + h_i \leq H. \end{cases} \quad (4)$$

In the equation, x_i , y_i , and z_i are the coordinate points of the i th cargo placed in the loading plan; the length of the cargo is l_i , the width is w_i , and the height is h_i , where $i = 1, 2, \dots, n$. It can be seen that the placement direction of the cargos is determined by these six state values.

2.3. Objective Function. In this paper, the average value of the inverse of the space utilization of all boxes is chosen as the optimization objective of the mathematical model of the three-dimensional offline box packing problem. The objective value of the three-dimensional offline bin packing

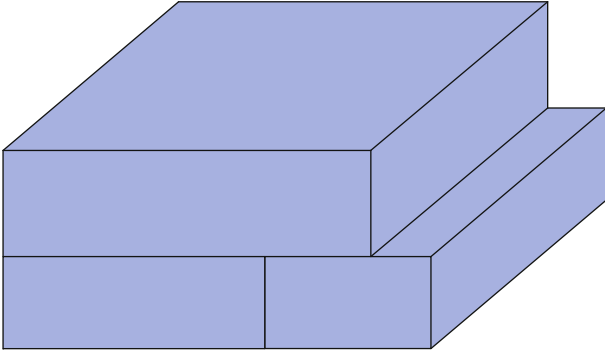


FIGURE 2: Schematic diagram of full support.

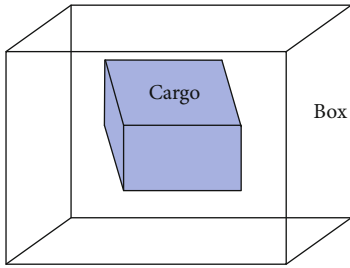


FIGURE 3: Schematic diagram of cargo crossed with box.

problem is

$$\begin{cases} Z = \sum_{j=1}^N \frac{V}{N \sum_{k=1}^{\text{num}_j} v_k}, \\ \min (Z). \end{cases} \quad (5)$$

In the equation, v_i is the volume of the i th cargo, and V is the volume of the box, N is the number of bins used in the packing scheme, and num_j represents the number of cargos loaded in the j th box in the loading scheme.

3. HGA for 3D Offline Binning Problem

This paper proposes a HGA that combines the genetic algorithm and the tabu search algorithm to solve the three-dimensional offline bin packing problem, as shown in Figure 4. This paper also designs a corner-occupancy heuristic loading algorithm (COHLA) based on the division and merging of the remaining space, which converts the solution in the intelligent search algorithm into a packing scheme that conforms to the mathematical model. After combining the hybrid genetic algorithm with the heuristic loading algorithm, the HGA for solving the three-dimensional offline bin packing problem is formed.

HGA is designed under the basic framework of genetic algorithm, including coding method, population initialization, fitness value calculation, selection, crossover and mutation, and other processes. This paper makes corresponding changes to the three-dimensional offline bin packing problem, designs a two-stage real number encoding method

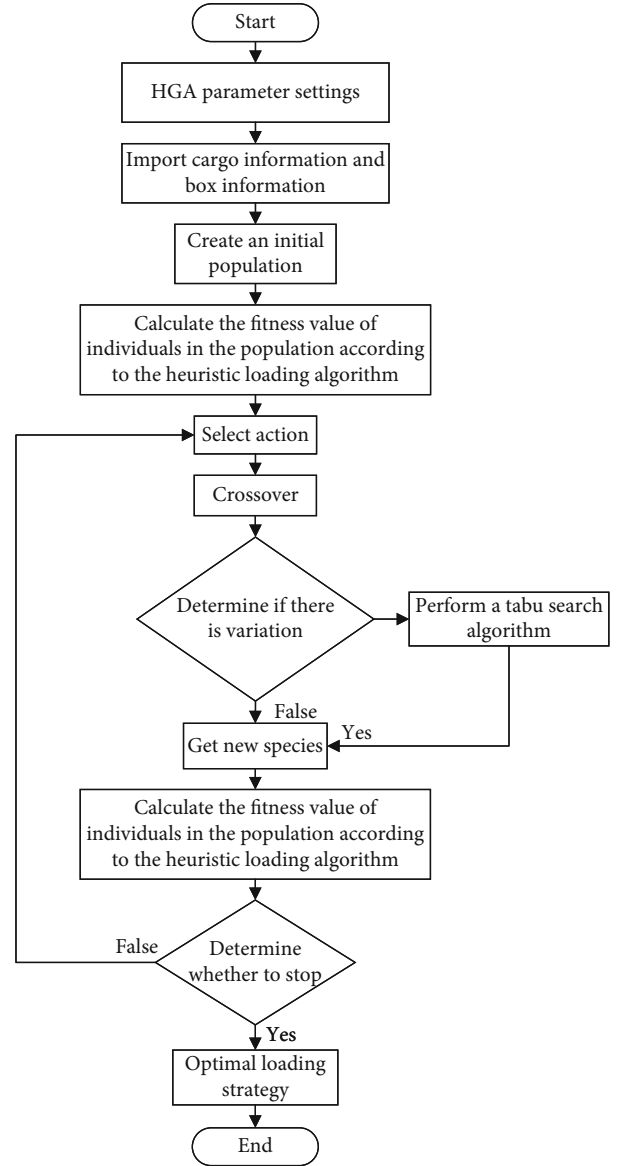


FIGURE 4: Flowchart of the HGA.

(TSRNEM) based on random keys, calculates the fitness value according to the heuristic loading algorithm, and chooses to use the roulette method and the best preservation method. For the hybrid selection operation of the strategy, a crossover operation based on partial random key and uniform crossover is designed, and the tabu search algorithm is used as the mutation operation.

3.1. Encoding Method. Common coding methods in genetic algorithms include binary coding, Gray code coding, real number coding, and integer or alphabetical arrangement coding [24]. Aiming at the three-dimensional offline bin packing problem, this paper designs a TSRNEM based on random keys as the encoding method.

For the packing problem of n -cargos to be loaded, the two-segment real number code contains real $n + n$ numbers, each real number is a random number between 0 and 1, and

each real number has nothing to do with the box serial number and box type. The first n real numbers are the packing sequence S , in which the i th real number s_i represents the packing sequence value of the cargos with the item number i ; the second n real numbers are the rotation state R , of which the i th real number r_i represents the rotation state of the cargos with the item number i .

As shown in Table 1, to get the packing order of the cargos, the algorithm will sort the cargo numbers according to the size of each real value in the packing sequence, and the resulting sequence of cargos numbers is the rotation state. For cuboid cargos and cuboid boxes, there are six relative rotation states between the two, and each state is marked with 1 to 6. In this algorithm, each cargo data contains a rotation state table T_i that records the allowed rotation state of the cargo. In order to meet the rotation constraints and obtain the rotation state of the cargos when they are packed, this paper multiplies the rotation state r_i of the cargos number by the rotation state table size T_i contained in the corresponding cargos and rounds up to obtain an integer T_{ij} ; the T_{ij} th rotation state in the rotation state table of the corresponding cargos is the rotation state of the cargos. As shown in Table 2, the packing state value r_2 of cargo 2 is 0.66, and the rotation state table T_2 in cargo 2 is 1, 2, 4, and 6, and the size T_{2s} of the rotation state table is 4. According to equation (6), the value of T_{2s} can be obtained as 3, the rotation state of the cargo 2 is the third rotation state in the rotation state table T_2 , which is the rotation state 4, and the corresponding rotation state of the cargo 2 in this packing scheme is $(x - w, y - h, z - l)$.

$$\begin{cases} Z = \sum_{j=1}^N \frac{V}{N \sum_{k=1}^{\text{num}_j} v_k}, \\ \min(Z). \end{cases} \quad (6)$$

3.2. Heuristic Loading Algorithm. In the genetic algorithm, in order to evaluate the pros and cons of individuals in the population, the objective function of the problem is converted into a fitness function, and the fitness function value of each individual represents the survival probability of the individual. At the same time, by decoding individual chromosomes, only the order and placement direction of the cargos can be obtained. To obtain an effective loading plan and fitness value, it is necessary to design an appropriate loading algorithm.

A good loading algorithm needs to consider positioning rules, space selection rules, space division rules, and remaining space merging rules [25]. In this paper, the COHLA based on remaining space division and merging is designed.

3.2.1. Positioning Rules and Spatial Selection Rules. In the process that the cargos are loaded into the boxes one by one according to a certain loading order, there should be a suitable positioning rule to clarify the placement position of each cargos in the box space. In order to meet the stability requirements and improve the utilization of space, this paper chooses to use the corner occupation strategy as the positioning rule (as shown in Figure 5). The basic idea of the corner strategy is as follows: first, place the cargos in a corner of

TABLE 1: The TSRNEM process (before).

Numbers	1	2	3	...	n
Packing sequence	0.11	0.10	0.51	...	0.44
Rotation state	0.58	0.94	0.14	...	0.88

the box, then place it along the edge of the box, and finally fill the box.

In each cuboid space, there are 8 corner areas, namely, the lower left corner, the lower left corner, the lower right corner, the lower right rear corner, the upper left corner, the upper left corner, the upper right corner, and the upper right rear corner. In order to meet the stability of the cargos, the cargos are selected to be placed in the lower four corners of the space; considering the actual loading process, the cargos are selected to be placed in the lower left corner and the lower right corner of the space. Due to the division of the remaining space, the bottom surface of each remaining space is the top surface of other cargos or the bottom surface of the box, which can ensure that the new cargos have support and ensure stability constraints. Therefore, this paper chooses the left rear lower corner rule as the positioning rule, that is, placing the cargos in the left rear lower corner in the space [26].

After constant refilling, there may be scattered remaining spaces in the box. Therefore, when a certain cargo to be loaded is to be loaded into the box, it is necessary to first select the available remaining space from all the remaining spaces and then select a remaining space from the available remaining spaces as the loading space of the cargo.

First, based on the size of the cargo, how it is rotated, its weight, and the size of each remaining space, the remaining space that can be fully loaded is picked.

For example, the length, width, height, and weight of the cargos are l_b , w_b , h_b , and m_b ; the rotation state is $(x - w, y - h, z - l)$; the length, width, and height of a certain remaining space are L , W , and H , respectively; and the total weight of the loaded cargos in the box is m_s ; if it satisfies equation (7), the size of the remaining space is larger than the size of the cargos and does not exceed the upper limit of the load of the box; the cargos can be loaded into the remaining space; otherwise, the remaining space is unavailable space. Then, to reduce the division of large space and improve the utilization rate of space, this paper selects the space with the smallest volume from the remaining spaces that can be loaded as the loading space.

$$\begin{cases} L \geq w_b, \\ W \geq h_b, \\ H \geq l_b, \\ m_s + m_b \leq M. \end{cases} \quad (7)$$

3.2.2. Remaining Space Division Rules. Remaining space refers to the space in the box that is not occupied by cargos. To ensure the stability of the cargos in the box, when the cargos are in the box, the cargos must be placed in parallel or orthogonal to the coordinate axis of the container, and no

TABLE 2: The TSRNEM process (after).

Numbers	1	2	3	4	5
Packing sequence	0.20	0.17	0.64	0.88	0.34
Rotation state	0.58	0.66	0.72	0.25	0.37
Numbers	2	1	5	3	4
Packing sequence	0.17	0.20	0.34	0.64	0.88
Rotation state	0.66	0.58	0.37	0.72	0.25

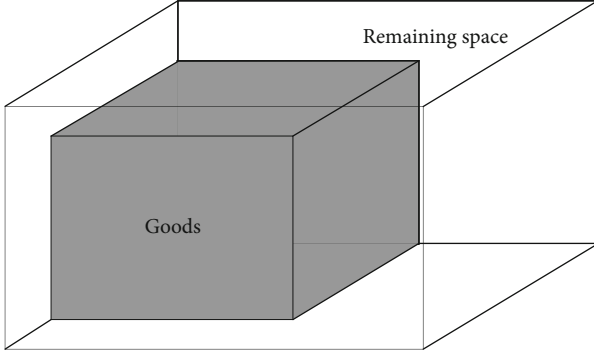


FIGURE 5: Schematic diagram of cargo placement.

inclined placement is allowed. Therefore, after the cargos are loaded into the box for many times, the phenomenon of cutting the remaining space in the box into complex irregular-shaped spaces occurs. For a longer cuboid space, it is more difficult to select the position of the cargos in the complex irregular shape space, and the amount of calculation required to complete the loading is larger.

To solve the complex irregular-shaped space problem, this paper chooses to divide the space when the first cargos are loaded into the box. When a cargo is loaded into the cuboid space, the cuboid space can be divided into four parts, namely, the space occupied by the cargo, the upper space, the front space, and the right space. Due to stability constraints and positioning rules for the lower left rear corner, the cargo can only be placed on the bottom of the box or on the cargo already packed in the box. Therefore, the upper space can only be the space directly above the cargos, and there are two division methods for the right space and the front space, such as vertical division and horizontal division. During the loading process, it is not desirable that the box space is divided into a large number of narrow and long thin strip-shaped residual spaces. Therefore, in the process of dividing the remaining space, the judgment will be made according to equation (8). If the value d is greater than 0, the vertical split is selected; otherwise, the horizontal split is selected.

$$\begin{cases} L \geq w_b, \\ W \geq h_b, \\ H \geq l_b, \\ m_s + m_b \leq M. \end{cases} \quad (8)$$

In equation (8), $s.l$ and $s.w$ represent the length and width of the remaining space, respectively, $g.l$ and $g.w$ represent the length and width of the cargo, respectively, and d represents the difference between the bottom area of the right space divided vertically and the bottom area of the front space divided horizontally.

3.2.3. Remaining Space Merge Rules. The residual space designed in this paper for the residual space merging operation is the abandoned space, the small space that cannot be put into any cargo. During the remaining space merging operation, the two spaces being operated should meet the following requirements before proceeding:

- (1) The two abandoned spaces are adjacent, adjacent to left and right or adjacent to front and rear
- (2) The two abandoned spaces need to have one side coplanar
- (3) The size of the new space generated after being merged is larger than the size of the original space
- (4) According to the merging conditions of the abandoned space, there are two merging methods: front and rear merging (as shown in Figure 6) and left and right merging (as shown in Figure 7). In order to make full use of the space in the box, after each item is put in, the space merging operation is performed on all the abandoned spaces until it cannot be merged

3.2.4. Heuristic Loading Algorithm Flow. In this paper, a COHLA based on the division and merging of the remaining space is designed (as shown in Figure 8); the heuristic loading algorithm will maintain a space list for each box. The list records the length, width, height, coordinates of the lower left corner, and other information of each remaining space in the box. The initial remaining space in the list is the box.

3.3. Selection, Crossover, and Mutation Operations. The HGA proposed in this paper is developed based on the framework of genetic algorithm, and selection, crossover, and mutation are the main processes in genetic algorithm. Aiming at the three-dimensional offline bin packing problem and the TSRNEM based on random keys, this paper chooses to use the roulette method and the mixed selection operation of the best preservation strategy, designs a crossover based on random and uniform crossover of partial keys operation, and uses the tabu search algorithm as the mutation operation.

3.3.1. Select Operation. In the genetic algorithm, the selection operation embodies the idea of survival of the fittest and survival of the fittest in the theory of evolution and simulates the process of species selection in nature. The selection operation is to screen out a part of individuals from the parent generation to generate offspring. The higher the fitness value of the individual is, the greater the probability of being selected. On the contrary, the lower the fitness value of the individual is, the lower the probability of being selected.

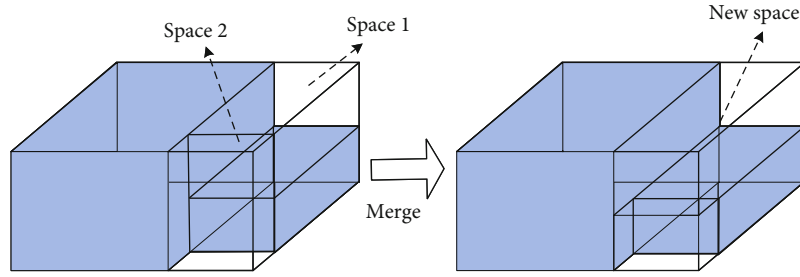


FIGURE 6: Schematic diagram of before and after merging of abandoned spaces.

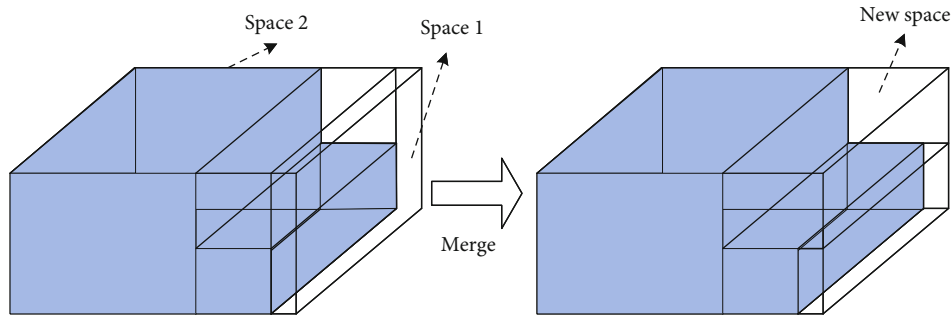


FIGURE 7: Schematic diagram of left and right merging of abandoned spaces.

The selection operation can make the population develop in the direction of higher fitness. In this process, the excellent genes and performance are preserved; it can ensure that the population can converge to the optimal solution range during the search process, but the selection operation method is not suitable. Now, there are the following 3 ways of selection operation that are proposed and verified.

The roulette method is based on the fitness value f_i of each individual and the sum $f = \sum f_i$ of the fitness values of all individuals and can obtain the probability $P_i = f_i/f$ of each individual being selected; in the selection process, the program will automatically generate a random value λ between 0 and 1. If there is an individual i which satisfies $P_1 + P_1 + \dots + P_i \leq \lambda < P_1 + P_1 + \dots + P_i + P_{i+1}$, the individual i is selected to enter the next link. The roulette method gives each individual a chance to be selected, and it has the characteristics of random sampling and has better global search ability, but it also leads to a decrease in the convergence speed of the algorithm and even makes it impossible to find the optimal solution.

The optimal preservation strategy is to sort all individuals in the population according to the fitness value from high to low and select the top individuals from the sorting to enter the next link, while other individuals will be eliminated. This method can ensure that the genes of the optimal individual will be inherited, but if the chromosomes of multiple individuals in the selected K individuals are consistent, it will easily lead to a local optimal solution, so that the algorithm cannot find the global optimal solution untie.

Tournament method randomly selects a certain amount of individuals from the parent generation, selects the individual with the best fitness value from this amount of individuals into the offspring, and repeats the process until the

selection is full. The tournament method will be both selective and deterministic, a method between the roulette method and the optimal preservation strategy.

In this paper, the parent individuals selected in the selection operation are directly regarded as the offspring individuals. Moreover, in the early stage of the genetic algorithm, when the iteration g is less than the predetermined value g_0 , the roulette method is selected as the selection operation method, so that the search can be carried out in a wider range and the global search ability is maintained; in the later stage, when the iteration g is greater than the predetermined value g_0 , the optimal preservation strategy is used as the selection method. The operation method enhances the search ability of the algorithm in the optimal solution range and accelerates the convergence speed.

$$g_0 = G \times \frac{2}{3}. \quad (9)$$

In equation (9), G is the number of iterations.

3.3.2. Crossover Operation. In the genetic algorithm, the crossover operation simulates the genetic recombination process in genetics, which reflects the characteristics of gene exchange in biology. It is an important operation in the genetic algorithm and the main way to form a new individual in the genetic algorithm. Crossover operation is to take the individuals in the parent as parents and exchange the genes in the chromosomes of the parents according to certain rules, so as to generate a new individual with a combination of traits of the two parents. Commonly used crossover

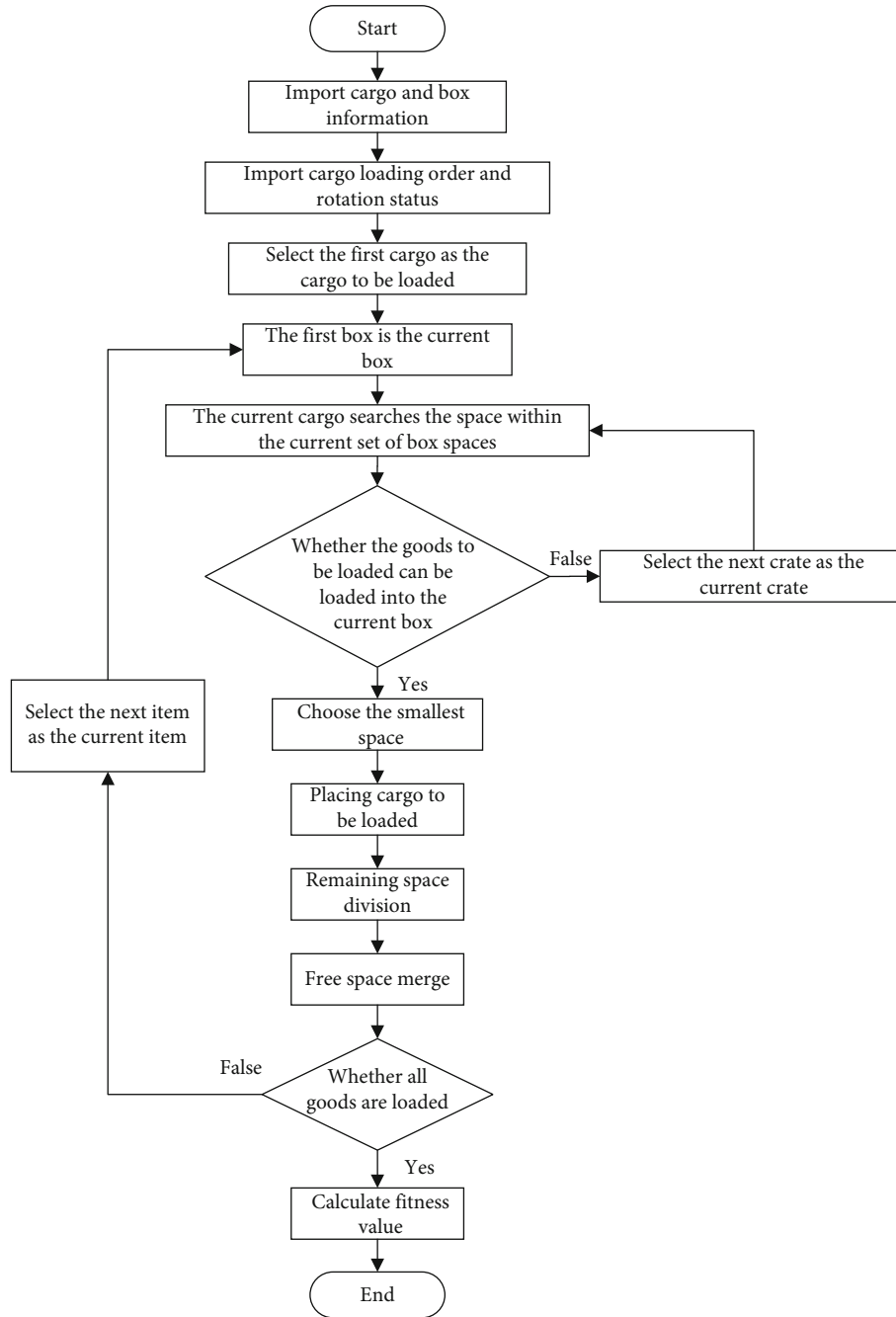


FIGURE 8: Flowchart of heuristic loading algorithm.

methods include single-point crossover, multipoint crossover, and uniform crossover.

Aiming at the TSRNEM based on random key, this paper designs a crossover operation method based on partial random key and uniform crossover. In this paper, we first choose to use two individuals to generate child individuals through crossover operation. These two individuals are called parent individuals. One of the parent individuals is derived from the individuals selected by the selection operation, and the other is randomly selected from the parent population. Compared with other methods, this method is easier to converge and can improve the quality of individuals

in the population; then, along the gene locus in the chromosome, select the gene from the chromosomes of the two parent individuals as the gene of the offspring at that locus; the selection method is to generate a random number between 0 and 1. If the random number is less than the crossover probability, the gene in the chromosome of the first parent individual is selected; otherwise, the gene in the chromosome of the second parent individual is selected.

As shown in Figure 9, chromosome 1 is derived from individuals randomly selected in the selection operation, and chromosome 2 is derived from randomly selected individuals in the parent population. Assuming that the

crossover probability is 0.65, for the first gene, the random number generated by the uniform distribution as the probability distribution function is 0.44, which is less than the crossover probability, so the first gene selection of the offspring is inherited from the first gene of chromosome 1. In the second gene, the random number is 0.72, which is greater than the crossover probability, so the second gene of the offspring is selected to inherit the second gene of chromosome 2. The latter genes in the progeny chromosomes are also generated according to this method.

3.3.3. Mutation Operation. In genetics, genes mutate in the process of inheritance, producing new genes, which in turn give individuals new shapes. In the genetic algorithm, mutation operation is to simulate the phenomenon of gene mutation in genetics, which is to change a certain segment of genes in an individual's chromosomes through a certain mutation method to generate a new individual. Although it is the same as the crossover operation to generate new individuals, the mutation operation is mainly to prevent and avoid the phenomenon of local optimum or precociousness and to ensure the diversity of the population and the auxiliary crossover operation to ensure the global search ability of the algorithm. At the same time, local search within the range of excellent solutions can be realized.

In genetic algorithm, mutation probability affects the performance of the algorithm. Although the mutation operation can generate new individuals, it cannot guarantee the pros and cons of the generated new individuals. If the mutation probability value is relatively large, the probability of the mutation operation to generate new individuals increases, which expands the global search ability of the algorithm but, at the same time, reduces the convergence ability of the algorithm. If the mutation probability value is relatively small, the probability of the mutation operation to generate new individuals decreases, which reduces the global search ability of the algorithm, and the phenomenon of precociousness or local optimization is prone to occur.

Genetic algorithm has the characteristics of weak local search ability, while tabu search algorithm can be used as a local search algorithm with strong local search ability, short solution speed, and fast convergence speed. Therefore, the tabu search algorithm is selected as the mutation operation of the genetic algorithm.

In this paper, mutation will be performed on all individuals in the offspring population produced by selection and crossover operations. The mutation operation process is as follows: first, for individual *A*, a random number between 0 and 1 is randomly generated, and the random number conforms to the average distribution. If the random number is less than the mutation probability, individual *A* will proceed to the next step; otherwise, keep individual *A* and perform mutation operation on the next individual; then, the chromosome of individual *A* is used as the initial solution of the tabu search algorithm, and the tabu search is performed; then, the optimal solution obtained through the tabu search is used as the chromosome to obtain the individual *B*, and the individual *B* is used to replace *A*; all individuals perform the previous steps and finally obtain a new

population after mutation and continue to perform the genetic algorithm-related steps on the new population.

4. Test Practice

In order to test the HGA proposed in this paper to solve the three-dimensional offline packing problem, this paper uses the classical test data and actual container packing data to test the proposed algorithm. The proposed HGA is implemented on MATLAB and realizes three-dimensional stereo display. The proposed algorithm runs in the following configuration environment:

Running software: MATLAB R2014a

Operating system: Windows 7 64 bit

Processor: Intel fourth-generation Core i5-4200 U dual-core

Memory: Hynix 8G

Hard disk: Toshiba MQ01ABF505 (500 GB/5400 rpm)

The proposed HGA includes parameters such as population size, iteration times, crossover probability, mutation probability, candidate solution set size, and tabu list length. Through multiple simulation verifications, this paper chooses to set the population size to 100, the number of iterations to 100, the number of elites to 50, the crossover probability to be 0.7, the mutation probability to be 0.1, the candidate solution size to be 30, and the tabu length to be 100.

4.1. Testing Based on Classic Test Data. To test the proposed algorithm, this paper adopts the BR data set recognized by domestic and foreign scholars. The BR dataset was proposed by Bischoff in 1995 [27, 28], which contains 15 sets of data, namely, BR1-BR15; each set of data has 100 test data, and with the increase of the set number, the variety increases and the heterogeneity of data increases.

Among them, BR1-BR7 are weakly heterogeneous datasets: the cargo data consists of only a few different cargo types, BR8-BR15 are strongly heterogeneous datasets: there are many different cargo types in the cargo data, but each type is only a small amount of individual cargos. For convenience, the data in the first group of data is denoted as BR i -1; for example, the first data in the third group of data is BR3-1.

Each set of data includes a box and its size, as well as a variety of cargos of different sizes, and records the size, quantity, and supported rotation of each type of cargos. As shown in Table 3, the size of the box in the BR3-1 data is $587 \times 233 \times 220$, and there are 8 kinds of cargos, the sizes of the 8 kinds of cargos are different, and there are respective quantities of cargos and supported rotation methods.

However, the BR dataset is designed for the single box loading problem with the goal of maximizing the output, and it is used to test the performance of related algorithms applicable to this problem. Therefore, in order to make the BR dataset suitable for the multibox feature in the 3D offline packing problem proposed in this paper, the test is performed after the number of cargos in the dataset is changed to 5 times the original number and compared with the

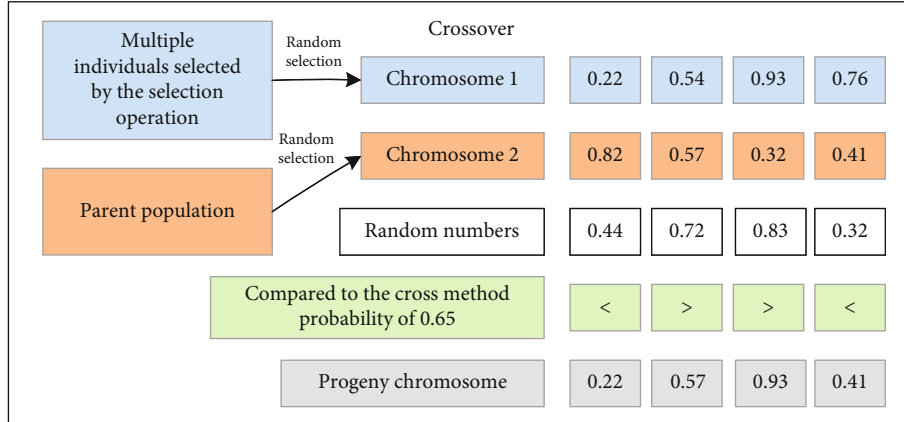


FIGURE 9: Schematic diagram of crossover operation.

TABLE 3: The information of box and cargo in BR3-1 data.

	Length	Width	Height	Quantity	Rotating state
Box	587	233	220		
Cargo 1	108	76	30	24	1, 3
Cargo 2	110	43	25	9	1, 2, 3, 5
Cargo 3	92	81	55	8	1, 2, 3, 4, 5, 6
Cargo 4	81	33	28	11	1, 2, 3, 5
Cargo 5	120	99	73	11	1, 2, 3, 4, 5, 6
Cargo 6	111	70	48	10	1, 2, 3, 5
Cargo 7	98	72	46	12	1, 2, 3, 5
Cargo 8	95	66	31	9	1, 3

TABLE 4: The test results of 4 algorithms.

Algorithm	Number of boxes required		
	BR1-1	BR3-1	BR5-10
Proposed algorithm	7	6	7
Genetic algorithm	7	7	8
Genetic simulated annealing algorithm	7	6	7
Hierarchical heuristics algorithm	8	7	8

genetic algorithm, genetic simulated annealing algorithm, hierarchical heuristic algorithm, and other algorithms.

4.2. *Testing Based on Weakly Heterogeneous Data.* BR1-BR7 are data with weak heterogeneity in the BR dataset. Therefore, in the testing process based on weak heterogeneous data, this paper chooses to use BR1-1, BR3-1, and BR5-10 in the BR dataset as the test data. Among them, there are 3 cargo types in BR1-1, 8 cargo types in BR3-1, and 12 cargo types in BR5-10. And when the number of cargos in the three data becomes 5 times the original data, it is compared with the genetic algorithm, genetic simulated annealing algorithm, hierarchical heuristic algorithm, and other algorithms [29–31].

The test results are shown in Tables 4 and 5. In the test of BR1-1 under 5 times, the algorithm proposed in this paper, the genetic algorithm, and the genetic simulated annealing algorithm can all fully load all the cargos in 7 boxes, and the hierarchical heuristic algorithm needs 8 boxes, and in the packing scheme obtained by the algorithm proposed in this paper, the space utilization rate of other boxes except no. 7 box is above 69%, and the highest is 85.6%, while the space utilization rate of the boxes in the other three algorithms is lower than that proposed algorithm in this paper. In the test of BR3-1 at 5 times, the algorithm proposed in this paper and the genetic simulated annealing algorithm can all load all the cargos in 6 boxes, while the genetic algorithm and the hierarchical heuristic algorithm need 7 boxes. In the packing scheme obtained by the proposed algorithm, the space utilization rate of other boxes except box 6 is above 72%, and the highest is 83.2%, which is better than the other three algorithms. In the test of BR5-10 under 5 times, the algorithm proposed in this paper and the genetic simulated annealing algorithm can all load all the cargos in 7 boxes, while the genetic algorithm and the hierarchical heuristic algorithm need 8 boxes. In the packing scheme obtained by the proposed algorithm, the space utilization rate of other boxes except box 7 is above 74%, and the highest is 85.3%, which is better than the other three algorithms.

To sum up, in the test of weak heterogeneous data, the HGA proposed in this paper is not only better than other algorithms in the number of boxes but also better than other algorithms in the space utilization of boxes, and the stronger the heterogeneity is, the better the packing scheme is.

4.3. *Testing Based on Strongly Heterogeneous Data.* In the testing process based on strongly heterogeneous data, this paper chooses to use BR8-1, BR9-1, and BR10-1 in the BR dataset as the test data. Among them, there are 30 types of cargos in BR8-1, 40 types of cargos in BR9-1, and 50 types of cargos in BR10-1. And in the case that the number of cargos in the three data becomes 5 times, it is compared with the genetic algorithm, genetic simulated annealing algorithm, hierarchical heuristic algorithm, and other algorithms.

TABLE 5: The space utilization of each box in the packing scheme obtained by the 4 algorithms.

Type of data	Algorithm	The space utilization of each box (%)							
		1	2	3	4	5	6	7	8
BR1-1	Proposed algorithm	85.6	80.3	79.4	76.6	72.3	69.3	30.5	
	Genetic algorithm	72.3	74.6	65.8	66.9	69.2	71.7	73.6	15.9
	Genetic simulated annealing algorithm	74.4	71.7	73.3	68.3	72.3	69.4	65.7	
	Hierarchical heuristics algorithm	60.3	72.3	64.3	65.8	70.8	66.7	69.3	
BR3-1	Proposed algorithm	83.2	73.6	79.2	72.3	72.3	41.3		
	Genetic algorithm	63.3	60.7	59.9	62.3	60.3	63.9	60.1	
	Genetic simulated annealing algorithm	73.6	72.9	62.2	72.3	70.0	70.8		
	Hierarchical heuristics algorithm	65.3	67.9	59.1	70.2	68.3	68.7	21.3	
BR5-10	Proposed algorithm	85.3	83.9	80.3	81.3	76.4	74.5	12.3	58.4
	Genetic algorithm	64.3	61.7	58.2	65.3	61.2	62.8	61.1	
	Genetic simulated annealing algorithm	72.3	70.6	69.7	73.4	69.3	68.8	67.0	30.6
	Hierarchical heuristics algorithm	63.5	69.2	66.9	64.1	68.3	64.4	65.9	

The test results of the three groups of strongly heterogeneous test data are shown in Tables 6 and 7. In the test of BR8-1 at 5 times, both the algorithm proposed in this paper and the genetic simulated annealing algorithm can fully load all the cargos in 7 boxes. The packing scheme obtained by the genetic algorithm and the hierarchical heuristic algorithm needs to be 8 boxes, and the space utilization rate of the other boxes in the packing scheme obtained by the algorithm proposed in this paper is more than 78%, and the highest is 84.6%, while the space utilization rate of the boxes in the other three algorithms is lower than the algorithm proposed in this paper. In the test of BR9-1 under 5 times, the algorithm proposed in this paper can fully load all the cargos with 7 boxes, while the other three algorithms need 8 boxes. In the scheme, the space utilization rate of other boxes except box 7 is above 74%, and the highest is 85.7%, which is better than the other three algorithms. In the test of BR10-1 under 5 times, the algorithm proposed in this paper can fully load all the cargos with 7 boxes, but the other three algorithms need 8 boxes. The space utilization rate of all boxes except box 7 in the scheme is above 70%, and the highest is 83.2%, which is better than the other three algorithms.

To sum up, the test results on strong heterogeneous data are similar to those on weak heterogeneous data. The HGA proposed in this paper is not only better than other algorithms in terms of the number of boxes but also better than other algorithms in terms of box space utilization.

4.4. Testing Based on Actual Packing Data. The actual packing case used in this paper is the actual packing data of a logistics transportation company. The company is mainly responsible for providing transportation services for enterprises in the industrial park and transporting cargos in the park to other urban logistics bases in the province. Since there are dozens of types and large quantities of cargos produced by enterprises in the park, the company needs to optimize the loading process of vehicles, reduce the number of

TABLE 6: The test results of 4 algorithms.

Algorithm	Number of boxes required		
	BR8-1	BR9-2	BR10-1
Proposed algorithm	7	7	7
Genetic algorithm	8	8	8
Genetic simulated annealing algorithm	7	8	8
Hierarchical heuristics algorithm	8	8	8

transport vehicles, reduce transport costs, and improve transport capacity.

Logistics transportation company A needs a cargo packing plan. The overall goal is to complete the transportation of all cargos with the least number of trucks and to use the least number of trucks to complete the loading of all cargos. At the same time, the volume constraints, guillotine constraints, and load constraints must be satisfied during the loading process, such as the cargo in the truck box cannot be operated with the height limit of the truck, and the total weight of the cargo in the truck cannot exceed the weight limit of the truck; the cargo cannot be divided.

Two of the service companies of logistics transportation company A are major customers, and they are replaced by B and C, respectively. Company B mainly produces 8 kinds of cargos, and company C mainly produces 12 kinds of cargos (the relevant parameters of cargos produced by companies B and C are shown in Table 8).

The logistics transportation company A mainly uses vans. The size of the vans is $8200 \times 2400 \times 2400$ mm, and the maximum load is 10000 kg. According to the actual transportation case of logistics transportation company A, this paper selects case A and case B as the test cases of this paper. Case A (as shown in Table 9) transports a batch of cargos produced by company B to other places. The cargo data in the case is cargos produced by company B; case B

TABLE 7: The space utilization of each box in the packing scheme obtained by the 4 algorithms.

Type of data	Algorithm	The space utilization of each box (%)							
		1	2	3	4	5	6	7	8
BR8-1	Proposed algorithm	84.6	82.3	80.4	78.6	79.8	78.8	9.5	
	Genetic algorithm	62.3	63.2	59.3	57.6	58.1	66.1	65.6	60.8
	Genetic simulated annealing algorithm	72.8	68.7	71.2	71.6	70.9	68.5	69.3	
	Hierarchical heuristics algorithm	69.7	65.1	68.4	64.6	64.1	65.3	63.5	32.3
BR9-1	Proposed algorithm	85.7	82.1	80.8	80.2	78.6	77.4	15.1	
	Genetic algorithm	66.4	64.9	62.2	63.9	58.7	62.8	61.8	58.3
	Genetic simulated annealing algorithm	65.6	63.4	58.6	64.2	59.1	64.1	61.7	62.3
	Hierarchical heuristics algorithm	70.3	67.9	62.1	65.2	65.3	68.7	62.2	37.3
BR10-1	Proposed algorithm	83.2	80.6	78.8	73.5	72.9	70.5	39.2	
	Genetic algorithm	70.9	63.7	59.7	62.3	58.1	57.6	60.4	66.0
	Genetic simulated annealing algorithm	64.5	68.9	57.6	59.8	62.9	57.9	63.9	63.2
	Hierarchical heuristics algorithm	72.6	69.9	68.4	63.1	62.8	60.7	60.5	40.7

TABLE 8: The cargos produced by companies B and C.

The company	Serial number	Number of the required boxes			Weight (kg)
		Length (mm)	Width (mm)	Height (mm)	
B	1	1100	800	550	2.0
	2	400	400	1600	6.0
	3	2000	1500	950	5.8
	4	1800	1200	1200	9.0
	5	1600	800	800	6.2
	6	600	600	800	3.2
	7	700	700	800	4.2
	8	800	800	800	3.6
C	1	1600	1200	1200	10.2
	2	800	950	750	2.5
	3	600	600	800	4.2
	4	800	900	500	4.3
	5	600	550	450	7.0
	6	1500	1200	1200	11.2
	7	650	500	800	3.8
	8	1000	1300	1000	7.6
	9	800	800	600	6.2
	10	800	600	400	4.8
	11	650	800	550	3.6
	12	850	600	600	5.8

TABLE 9: The data of cargos in case A.

The serial number	1	2	3	4	5	6	7	8
The quantity of required cargos	12	11	8	11	15	7	12	9

(as shown in Table 10) is to transport a batch of cargos produced by company C to other places, and the cargo data in the case are cargos produced by company C.

For case A, we use the optimization model and proposed hybrid genetic algorithm to optimize the packing. After optimization, it is obtained that 2 vehicles are required to load all the cargos in case A. In the original packing process based on workers' packing experience, three trucks are required to complete case A, and the space utilization rate of each truck is about 65%. As shown in Table 11, the space utilization rate of both vehicles is more than 80%, especially the space utilization rate of car 1 is more than 90% after using the proposed algorithm. The packing scheme obtained by the hybrid genetic algorithm reduces the use quantity of the truck by one and consequently reduces the number of trucks, fuel consumption, driver wages, maintenance, and other costs and improves the company's interests.

For case B, the optimization model and proposed hybrid genetic algorithm are used for packing optimization. After optimization, it is obtained that 3 vehicles are required to load all the cargos in case A. The space utilization rate of each vehicle in the packing plan is shown in Table 12. The compartments of vehicle 1 and vehicle 2 are filled with cargos, and their space utilization rate is more than 80%, especially the space utilization rate of vehicle 1 is 92.1%. The vehicle 3 has a small amount of cargos to be loaded, which leads a low level of the space utilization rate. The situation that the space utilization rate of vehicle 1 > the space utilization rate of vehicle 2 > the space utilization rate of vehicle 3 is due to the proposed hybrid genetic algorithm. The algorithm will give priority to loading the cargos into the front-most compartment, so the rear compartment may not be full, which will reduce the space utilization.

According to the survey of the packing workers, case B requires 4 to 5 vehicles to load in the packing process. There is a large waste of space in the compartment of each vehicle, and the space utilization rate is about 60%. Compared with the original scheme, the proposed hybrid genetic algorithm increases the space utilization rate of each vehicle, reduces the number of vehicles required, and greatly reduces the cost of logistics transportation company A.

TABLE 10: The data of cargos in case B.

The serial number	1	2	3	4	5	6	7	8	9	10	11	12
The quantity of required cargos	9	16	20	17	14	15	20	8	16	18	16	10

TABLE 11: The space utilization rate of each vehicle in the packing plan of case A.

The serial number	1	2
The space utilization rate (%)	90.2	80.3

TABLE 12: The space utilization rate of each vehicle in the packing plan of case B.

The serial number	1	2	3
The space utilization rate (%)	90.1	82.1	58.7

5. Conclusion

Aiming at the direction and stability of the loaded cargos, this paper designs a heuristic loading algorithm using the positioning rule of the left rear bottom corner. The HGA combining genetic algorithm and tabu search algorithm are designed to make up for the weak local search ability and slow convergence speed of genetic algorithm. In order to test the performance of the HGA, this paper uses 3 sets of weakly heterogeneous data and 3 sets of strong heterogeneous test data for testing. The algorithm can not only implement a packing scheme with a smaller number of boxes, but also more space utilization per box.

Data Availability

The data is available at <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 52177109 and 51707135, in part by the Hubei Province Key Research and Development Program Funded Projects (2020BAB109), and the Natural Science Foundation of Hubei Province under Grant 2019CFB211.

References

- [1] M. Fazili, U. Venkatadri, P. Cyrus, and M. Tajbakhsh, "Physical internet, conventional and hybrid logistic systems: a routing optimisation-based comparison using the Eastern Canada road network case study," *International Journal of Production Research*, vol. 55, no. 9, pp. 2703–2730, 2017.
- [2] R. Sridharan and S. Domnic, "Placement strategy for inter-communicating tasks of an elastic request in fog-cloud environment," *Scalable Computing-Practice and Experience*, vol. 20, no. 2, pp. 335–348, 2019.
- [3] M. Delorme, M. Iori, and S. Martello, "Bin packing and cutting stock problems: mathematical models and exact algorithms," *European Journal of Operational Research*, vol. 255, no. 1, pp. 1–20, 2016.
- [4] K. Kyungdaw, M. Ilkyeong, and W. Hongfeng, "A hybrid genetic algorithm with a new packing strategy for the three-dimensional bin packing problem," *Applied Mathematics and Computation*, vol. 219, no. 3, pp. 1287–1299, 2012.
- [5] H. Mhand, N. Stephane, and W. Lei, "Hybrid greedy heuristics based on linear programming for the three-dimensional single bin-size bin packing problem," *International Transactions in Operational Research*, vol. 21, no. 1, pp. 59–79, 2014.
- [6] K. Ali, C. Francois, and T. El-Ghazali, "New lower bounds for bin packing problems with conflicts," *European Journal of Operational*, vol. 206, no. 2, pp. 281–288, 2010.
- [7] M. Gendreau, G. Laporte, and F. Semet, "Heuristics and lower bounds for the bin packing problem with conflicts," *Computers and Operations Research*, vol. 31, no. 3, pp. 347–358, 2004.
- [8] D. Tansel and C. Ahmet, "Optimization of one-dimensional Bin Packing Problem with island parallel grouping genetic algorithms," *Computers and Industrial Engineering*, vol. 75, no. 2, pp. 176–186, 2014.
- [9] G. Kavita and D. Kusum, "Investigation of suitable perturbation rate scheme for spider monkey optimization algorithm," *Proceedings of Fifth International Conference on Soft Computing For Problem Solving*, pp. 839–850, Springer, Singapore, 2016.
- [10] A. Martinez-Sykora, M. C. So, C. S. M. Currie, C. Bayliss, and J. A. Bennell, "Optimizing pricing and packing of variable-sized cargo," *IMA Journal of Management Mathematics*, vol. 32, no. 3, pp. 303–327, 2021.
- [11] N. Shuhei, L. Chonho, and M. Tomohiro, "Solving 3D container loading problems using physics simulation for genetic algorithm evaluation," *IEICE Transactions on Information and Systems*, vol. E104.D, no. 11, pp. 1913–1922, 2021.
- [12] L. Xiao, L. Jia-min, and C. An-xi, "HAPE3D—a new constructive algorithm for the 3D irregular packing problem," *Frontiers of Information Technology and Electronic Engineering*, vol. 16, no. 5, pp. 380–390, 2015.
- [13] Z. Shihua, Z. Zejun, W. Lijin, and Z. Yiwen, "List-based simulated annealing algorithm with hybrid greedy repair and optimization operator for 0–1 knapsack problem," *IEEE Access*, vol. 6, pp. 54447–54458, 2018.
- [14] P. Jialiang, A. Abd El-Latif Ahmed, and L. Qiong, "Multimodal biometric authentication based on score level fusion of finger biometrics," *Optik International Journal for Light and Electron Optics*, vol. 125, no. 23, pp. 6891–6897, 2014.
- [15] Z. Wenbin, O. Wee-Chong, and L. Andrew, "The six elements to block-building approaches for the single container loading problem," *Applied Intelligence*, vol. 37, no. 3, pp. 431–445, 2012.

- [16] A. G. Ramos, S. Elsa, and J. F. Oliveira, "A new load balance methodology for container loading problem in road transportation," *European Journal of Operational Research*, vol. 266, no. 3, pp. 1140–1152, 2018.
- [17] K. Ahmed and O. Ender, "An iterated multi-stage selection hyper-heuristic," *European Journal of Operational Research*, vol. 250, no. 1, pp. 77–90, 2016.
- [18] A. Shahriar, K. Daniel, and K. Ahmed, "Combining Monte-Carlo and hyper-heuristic methods for the multi-mode resource- constrained multi-project scheduling problem," *Information Sciences*, vol. 373, pp. 476–498, 2016.
- [19] C. S. Nand, J. Donghwi, and L. H. Min, "An evolutionary algorithm based hyper-heuristic for the set packing problem," *Harmony Search and Nature Inspired Optimization Algorithms*, vol. 741, pp. 259–268, 2019.
- [20] W. Jingjing and L. Wang, "A cooperative memetic algorithm with feedback for the energy-aware distributed flow-shops with flexible assembly scheduling," *Computers and Industrial Engineering*, vol. 168, article 108126, 2022.
- [21] P. Zixiao, D. Lei, and L. Wang, "A knowledge-based two-population optimization algorithm for distributed energy-efficient parallel machines scheduling," *IEEE transactions on cybernetics*, vol. 52, no. 6, pp. 5051–5063, 2022.
- [22] Z. Fuqing, H. Zhang, and L. Wang, "A Pareto-based discrete Jaya algorithm for multiobjective carbon-efficient distributed blocking flow shop scheduling problem," *EEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1–10, 2022.
- [23] Z. Fuqing, S. Di, and L. Wang, "A hyperheuristic with Q-learning for the multiobjective energy-efficient distributed blocking flow shop scheduling problem," *IEEE Transactions on Cybernetics*, pp. 1–14, 2022.
- [24] A. A. Marcos, L. Quentin, and W. Louis, "A machine learning-based approximation of strong branching," *Inforns Journal on Computing*, vol. 29, no. 1, pp. 185–195, 2017.
- [25] A. Moura and J. F. Oliveira, "A GRASP approach to the container-loading problem," *IEEE Intelligent Systems*, vol. 20, no. 4, pp. 50–57, 2005.
- [26] P. C. Pinto, A. Nagele, M. Ä. Dejori, T. A. Runkler, and J. Ä. M. C. Sousa, "Using a local discovery ant algorithm for Bayesian network structure learning," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 767–779, 2009.
- [27] J. C. Bansal, H. Sharma, S. S. Jadon, and M. Clerc, "Spider monkey optimization algorithm for numerical optimization," *Memetic Computing*, vol. 6, no. 1, pp. 31–47, 2014.
- [28] E. E. Bischoff, "Three-dimensional packing of items with limited load bearing strength," *European Journal of Operational Research*, vol. 168, no. 3, pp. 952–966, 2006.
- [29] B. Gavish and H. Pirkul, "Efficient algorithms for solving multiconstraint zero-one knapsack problems to optimality," *Mathematical Programming*, vol. 31, no. 1, pp. 78–105, 1985.
- [30] X. Feng, I. Moon, and J. Shin, "Hybrid genetic algorithms for the three-dimensional multiple container packing problem," *Flexible Services and Manufacturing Journal*, vol. 27, no. 2, pp. 451–477, 2013.
- [31] T. G. Crainic, G. Perboli, and R. Tadei, "Extreme point-based heuristics for three-dimensional bin packing," *Inforns Journal on Computing*, vol. 20, no. 3, pp. 368–384, 2008.