

Solution to Solid Wood Board Cutting Stock Problem

Min Tang, Ying Liu ^{*}, Fenglong Ding  and Zhengguang Wang 

College of Mechanical and Electronic Engineering, Nanjing Forestry University, Nanjing 210037, China; tangmin@njfu.edu.cn (M.T.); dfl@njfu.edu.cn (F.D.); nanlinwzg@njfu.edu.cn (Z.W.)

* Correspondence: liuying@njfu.edu.cn

Abstract: In the production process for wooden furniture, the raw material costs account for more than 50% of furniture costs, and the utilization rate of raw materials depends mainly on the layout scheme. Therefore, a reasonable layout is an important measure to reduce furniture costs. This paper investigates the solid wood board cutting stock problem (CSP) and establishes an optimization model, with the goal of the highest possible utilization rate for original boards. An ant colony-immune genetic algorithm (AC-IGA) is designed to solve this model. The solutions of the ant colony algorithm are used as the initial population of the immune genetic algorithm, and the optimal solution is obtained using the immune genetic algorithm after multiple iterations are transformed into the accumulation of global pheromones, which improves the search ability and ensures the solution quality. The layout process of the solid wood board is abstracted into the construction process of the solution. At the same time, in order to prevent premature convergence, several improved methods, such as a global pheromone hybrid update and adaptive crossover probability, are proposed. Comparative experiments are designed to verify the feasibility and effectiveness of the AC-IGA, and the experimental results show that the AC-IGA has better solution precision and global search ability compared with the ant colony algorithm (ACA), genetic algorithm (GA), grey wolf optimizer (GWO), and polar bear optimization (PBO). The utilization rate increased by more than 2.308%, which provides effective theoretical and methodological support for furniture enterprises to improve economic benefits.

Keywords: solid wood board; one-dimensional cutting stock problem; ACA; GA; immune system



Citation: Tang, M.; Liu, Y.; Ding, F.; Wang, Z. Solution to Solid Wood Board Cutting Stock Problem. *Appl. Sci.* **2021**, *11*, 7790. <https://doi.org/10.3390/app11177790>

Academic Editors: Giancarlo Mauri, Andrea Prati, Carlos A. Iglesias, Luis Javier García Villalba and Vincent A. Cicirello

Received: 31 July 2021

Accepted: 21 August 2021

Published: 24 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, with the rapid development of the social economy, the desire for, and expectation of, a high quality of life is becoming more common. Solid wood furniture is increasingly valued and favored by people because of its unique aesthetic feeling and superior material characteristics; the market demand for solid wood furniture is showing an upward trend year on year [1–3]. However, some countries have low forestation; in China, the forest coverage rate is only three-fifths of the world average, and the per capita forest area is less than one-quarter of the world average [4]. Solid wood board resources are scarce; this, together with the implementation of a forest cutting quota policy, has led to the supply of solid wood board in China being highly dependent on imports [5–8]. In addition, the production technology of most furniture enterprises remains outdated, which results in solid wood boards being only partially used. This waste results in increased production costs and the need to manage waste [9]. Therefore, measures must be taken to alleviate the divergence between supply and demand of solid wood boards by saving raw materials [10]. In 2020, 322 million pieces of wooden furniture were produced in China, with the average board consumption per piece of furniture being 0.6 m³. If the existing board utilization rate is increased by 1%, 193 million m³ boards can be saved; this shows that a slight increase in the utilization rate of raw materials can cause a significant decrease in the total consumption of raw materials. As the first step of the furniture manufacturing process, layout directly determines the utilization rate of raw materials [11–13]. Therefore, a

reasonable layout is an important measure to reduce material costs and improve economic benefits and is thus of great significance in promoting the sustainable development of the furniture manufacturing industry.

As shown in Figure 1, the solid wood board cutting stock problem (CSP) refers to placing the required standard boards of different lengths on the original boards of different lengths. Under the condition of meeting the requirements for the demand and size of standard boards, the used original boards are fully utilized to maximize the board utilization rate. In this paper, we only consider length, which thus belongs to the one-dimensional cutting stock problem. The one-dimensional cutting stock problem is a typical optimization problem. In terms of computational complexity, it is an NP-hard problem with the highest complexity; the solution time increases exponentially with the problem scale, and there is no accurate optimal solution in the polynomial time [14–16]. At present, most studies use heuristic methods to solve optimization problems, such as polar bear optimization [17–20], grey wolf optimizer [21–23], and genetic algorithm [24–26], which can find good approximate solutions in a limited time.

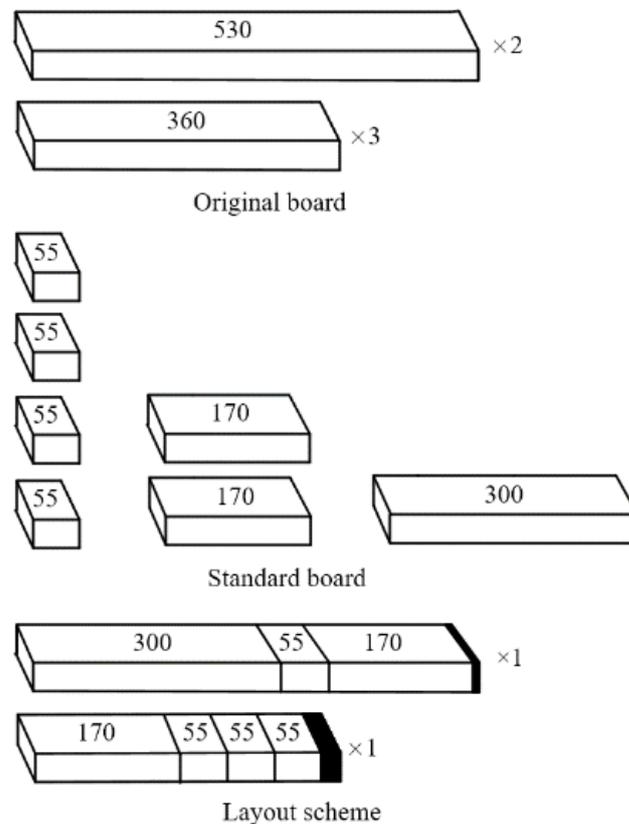


Figure 1. Solid wood board CSP.

The one-dimensional cutting stock problem has been studied by scholars for years. Common layout methods include column generation, greedy algorithms, and intelligent optimization algorithms. Sarper [27] proposed a priority-based goal programming approach to solve the one-dimensional cutting stock problem with a random demand mix. This methodology can evaluate the effects of various raw material order levels to account for demand randomness at the time of order. Cui [28] presented an integer programming model for the 1DCSPUL (one-dimensional cutting stock problem with usable leftovers) with limited leftover types, and described a heuristic algorithm based on a column-generation procedure for its solution. Cerqueira [29] proposed a modified greedy heuristic (MGH), which first orders the pair items, or the odd ones in the case that there are more items in the problem. The experimental results showed that the MGH is more effective than the

greedy heuristic. Campello [30] introduced the residual recombination heuristic (RRH) to solve the one-dimensional cutting stock problem and obtained high-quality solutions within a reasonable computational time. Ben [31] developed a solution method based on a particle swarm optimization (PSO) approach that takes into account the characteristics of the specific problem; this method was efficient for more than 78% of the total of the American Captain benchmark. In consideration of the characteristics of online detection, Lu [32] proposed a GPU parallel ant colony algorithm to solve the dynamic one-dimensional cutting stock problem; this algorithm had an obvious time efficiency advantage compared with the CPU traditional ant colony (CPUTAC) and branch and bound (BB).

Although the above algorithms are effective in helping to solve the one-dimensional cutting stock problem, it is difficult for a single algorithm to balance global search ability and local exploitation ability. A more advantageous improvement strategy is to combine two or more algorithms to make use of the advantages of a hybrid algorithm, making up for the defects of a single algorithm. Hua [33] proposed a global heuristic two-tier algorithm to solve the cable optimized cutting problem, which saved more than 3% of cable raw materials compared with traditional methods. Benjaoran [34] proposed a hybrid algorithm based on the intensive search algorithm (ISA), genetic algorithm (GA), and the best fit decreasing (BFD) algorithm. Test results showed that the hybrid-generated solutions were superior to the solutions received from the BFD algorithm alone. Anselmo [35] proposed a novel matheuristic algorithm based on a fix-and-optimize strategy hybridized with a random local search; this performed better than the CPLEX solver in larger instances, with an average relative percentage deviation (RPD) for objective values as high as 72%.

This paper takes the highest possible board utilization rate as the optimization goal and proposes an ant colony-immune genetic algorithm (AC-IGA) to solve the solid wood board cutting stock problem. In order to verify the feasibility and effectiveness of AC-IGA, it is compared with ant colony algorithm (ACA), genetic algorithm (GA), grey wolf optimizer (GWO), and polar bear optimization (PBO) through experiments.

2. Mathematical Model

In practical production, there are often different types of defects on the surface of solid wood boards; these have an impact on the structural strength and aesthetic degree. After obtaining the characteristic information of surface defects using machine vision technology, the defects are removed by the cutting saw, and multiple boards with different lengths are obtained [36–38]. These boards must be segmented according to the standard length for subsequent processing.

Assume that the total number of original boards after cutting defects is n . The length of the i -th original board is L_i ($i = 1, 2, \dots, n$). m types of standard boards need to be placed on these original boards without overlapping. The length and demand of the j -th type of standard board are l_j and b_j , respectively ($j = 1, 2, \dots, m$).

On the premise of meeting the constraints, the optimization goal is to minimize the total length of the used original boards, namely, to maximize the board utilization rate. The objective function is expressed as

$$\max U = \sum_{j=1}^m l_j b_j / L_C \quad (1)$$

where L_C is the total length of all the original boards selected in a layout scheme, U is the board utilization rate, and m is the number of types of standard boards.

The solid wood board CSP must meet the following three constraints:

(1) The total quantity of the j -th type of standard board placed on the i -th original board should be a nonnegative integer:

$$a_{ij} \in N \quad (2)$$

where a_{ij} is the total quantity of the j -th type of standard board placed on the i -th original board, and N is a nonnegative integer.

(2) The total quantity of the j -th type of standard board placed in a layout scheme is equal to its demand:

$$\sum_{i=1}^n a_{ij} = b_j \tag{3}$$

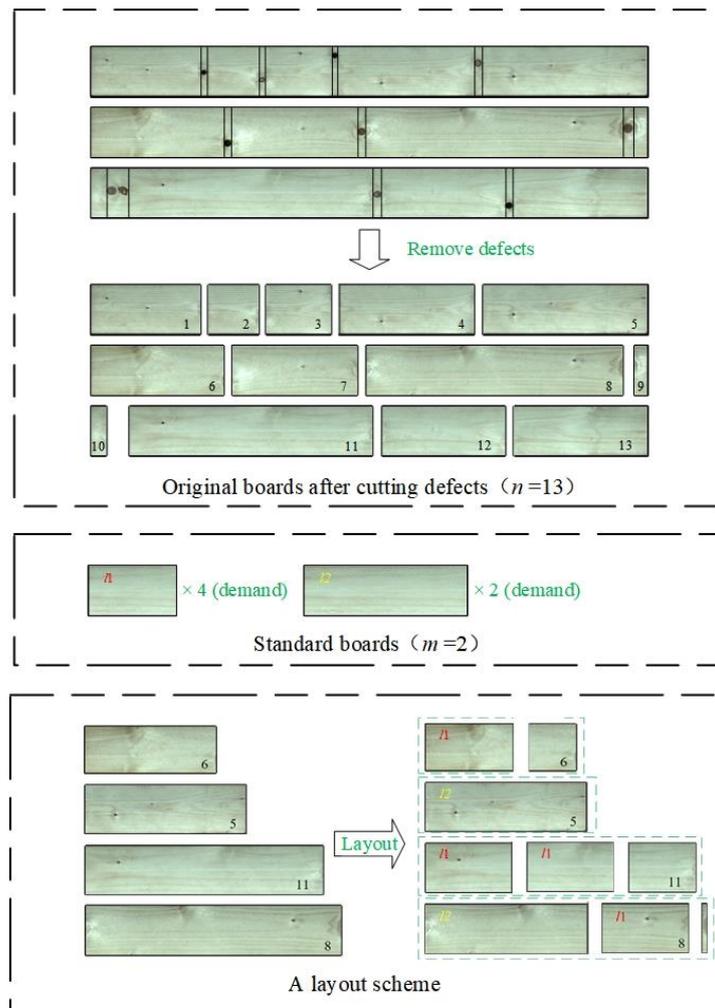
where b_j is the demand of the j -th type of standard board, and n is the total number of original boards.

(3) The total length of all the standard boards placed on the i -th original board cannot exceed the length of the i -th original board:

$$\sum_{j=1}^m a_{ij} l_j \leq L_i \tag{4}$$

where l_j is the length of the j -th type of standard board, and L_i is the length of the i -th original board.

Figure 2 shows the mathematical model of the solid wood board cutting stock problem. 5, 6, 8, and 11 are the selected original board numbers in a layout scheme.



$$U = \frac{l_1 \times 4 + l_2 \times 2}{L_c} \quad (L_c = L_5 + L_6 + L_8 + L_{11})$$

Figure 2. Mathematical model of the solid wood board cutting stock problem.

3. Algorithm Design

The solid wood board cutting stock problem is an NP-hard combinatorial optimization problem, and the quantity and length of boards are constrained, making this problem more complex. In the process of layout, not only should the original boards be selected, but the standard boards that can be placed on the selected original boards should also be selected. In addition, the selection quantity and length of the original boards are unknown, so the feasible solutions cannot be generated randomly.

The ant colony algorithm (ACA) is a typical algorithm for solving the combinatorial optimization problems. In the ACA, ants construct the feasible initial solutions through the design of pseudorandom state transition rule and taboo list when searching for paths, and continuously optimize paths through pheromone feedback mechanism. However, the ACA needs a long search time and is prone to premature stagnation when solving the large-scale problems.

The GA is a commonly used method in solving optimization problems, which has the global, fast search ability, but it does not use the feedback information in the system, which often leads to redundant iterations and low solving efficiency.

The AC-IGA is proposed to overcome the defects of the ACA and GA. In this paper, in order to maintain good population diversity and nonredundancy of information, the GA was improved according to the immune system, and the improved genetic algorithm was introduced into each iteration of the improved ant colony algorithm, so that the improved genetic algorithm can iterate in a good solution space and improve the search efficiency.

3.1. Important Concepts of the AC-IGA

3.1.1. Similarity

In this paper, the hamming distance refers to the number of different alleles between the chromosomes of two individuals and is used to measure similarity. The greater the number of different alleles, the greater the hamming distance and the smaller the similarity. Assuming that two individuals are numbered s and v , and each individual has B gene loci, the similarity between individual s and individual v is defined as follows:

$$S_{v,s} = \sum_{k=0}^{B-1} \eta_k / B \tag{5}$$

$$\eta_k = \begin{cases} 0, & s_k \neq v_k \\ 1, & s_k = v_k \end{cases} \tag{6}$$

where s_k is the k -th gene of individual s and v_k is the k -th gene of individual v .

Figure 3 shows the similarity between individual s and individual v , assuming that each individual has eight gene loci.

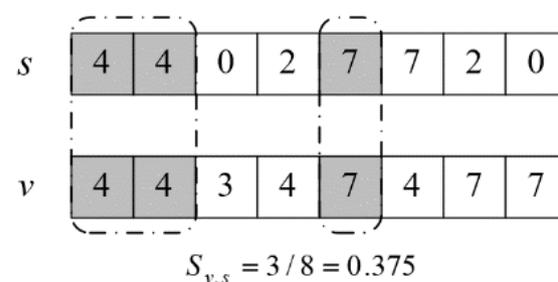


Figure 3. Similarity based on hamming distance.

3.1.2. Concentration

In this paper, concentration refers to the proportion of similar individuals in the population and represents the quality of population diversity [39]. In order to simplify the calculation, the definition of concentration based on information entropy is used for reference. The concentration of individual v is defined as

$$C_v = \frac{1}{N} \sum_{s=1}^N b_{v,s} \tag{7}$$

$$b_{v,s} = \begin{cases} 1, & S_{v,s} \geq Th \\ 0, & S_{v,s} < Th \end{cases} \tag{8}$$

where Th is the similarity threshold, $S_{v,s}$ is the similarity between individual s and individual v , N is the population size.

3.1.3. Pseudorandom State Transition Rule

The ant's selection of paths is mainly determined by pheromone and heuristic function [40]. In this paper, the probability of ant k transferring from the original board node i to the standard board node j at time t is calculated according to the pseudorandom state transition rule (shown in Equations (9) and (10)). When $q \leq q_0$, the standard board node is selected according to prior knowledge; otherwise, the standard board node is selected using the probability formula P_{ij}^k , as follows:

$$s = \begin{cases} \operatorname{argmax}([\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta) & q \leq q_0 \\ P_{ij}^k(t) & q > q_0 \end{cases} \tag{9}$$

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{r \in \text{allowed}_k} [\tau_{ir}(t)]^\alpha [\eta_{ir}(t)]^\beta} & j \in \text{allowed}_k \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where q is a random number uniformly distributed between $[0, 1]$, q_0 is the selection randomness parameter, α is the pheromone importance factor, β is the heuristic function factor, allowed_k is the set of optional standard board nodes, η_{ij} is the heuristic function, and τ_{ij} is the pheromone transferred from the original board node i to the standard board node j .

3.2. Algorithm Procedure

Figure 4 shows the flow chart of the AC-IGA. The IACA is framed by the red dash-dot line, IGA is framed by the blue dash-dot line, ANT_ITER is the maximum number of iterations of IACA, and IGA_ITER is the maximum number of iterations of IGA.

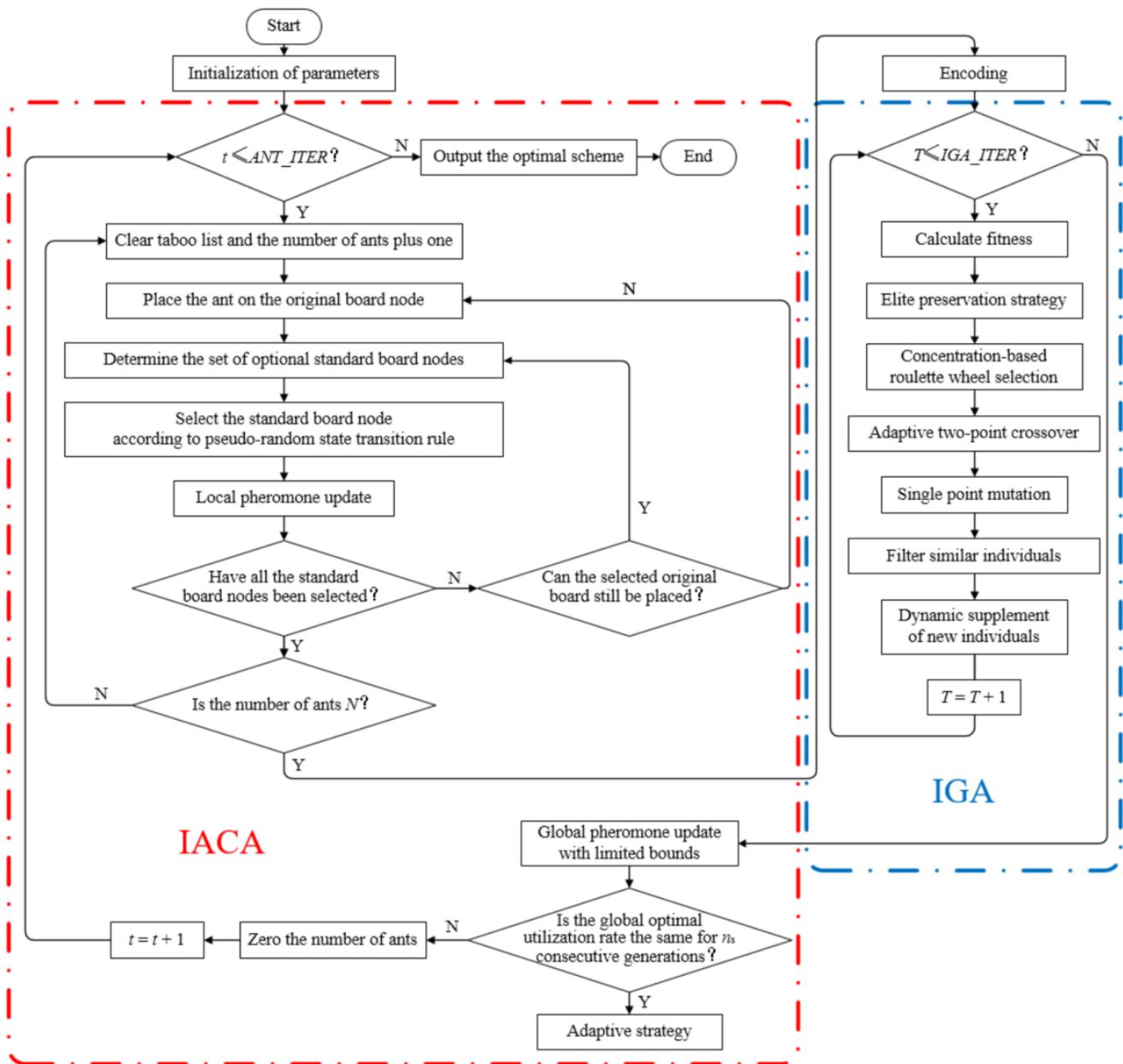


Figure 4. The flow chart of the AC-IGA.

3.3. Improved Ant Colony Algorithm (IACA)

3.3.1. Solution Construction

It is assumed that the total quantity of original boards is A , and the total quantity of standard boards is B . The original boards and standard boards are numbered sequentially from 0 to $A-1$ and $B-1$, respectively, and these numbers are used as nodes of the IACA. These nodes can be divided into original board nodes and standard board nodes; the original board node i represents the original board numbered i ($i \in [0, A-1]$) and the standard board node j represents the standard board numbered j ($j \in [0, B-1]$).

The layout process of solid wood boards can be described as follows. First, select an original board, and then select several standard boards for the layout to make the residual length of the selected original board as close to zero as possible. Then, replace the original board and repeat the above process until each standard board has been placed once. The concrete steps for ants to construct feasible solutions are as follows:

(1) $wcode$ is the set of optional original board nodes (the difference set between all original board nodes and $Tabu(x)$) and $scode$ is the set of optional standard board nodes (the difference set between all standard board nodes and $Tabu(y)$).

(2) Taking the length of the original board corresponding to the original board node i as L_i ($i \in wcode$) and the length of the standard board corresponding to the standard board node j as l_j ($j \in scode$), record the maximum lengths as L_{max} and l_{max} , respectively.

(3) Assuming that the original board node c is selected, the method of selecting the original board node is as follows:

$$c = \underset{i \in wcode}{argmin} [L_i \geq (l_{max} + r)] \tag{11}$$

where r is a random number uniformly distributed between $[0, L_{max} - l_{max}]$.

(4) Select the standard board corresponding to the standard board node in $scode$, one by one, to establish a tentative layout, and take the residual length of the selected original board after the tentative layout as a parameter of the heuristic function η_{cd} :

$$\eta_{cd} = \begin{cases} 1 / (R_c - l_d + 1) & R_c \geq l_d \\ 0 & R_c < l_d \end{cases} \tag{12}$$

where R_c is the residual length of the original board corresponding to the selected original board node c , and l_d is the length of the standard board corresponding to the standard board node d in the tentative layout ($d \in scode$).

In combination with the corresponding pheromone (the pheromone matrix is a two-dimensional array: the first dimension is the original board number, and the second dimension is the standard board number), select the standard board node according to the pseudorandom state transition rule.

(5) Record the selected original board node and the selected standard board node in the taboo list $Tabu(x)$ and $Tabu(y)$, respectively.

(6) Update $wcode$ and $scode$.

(7) If there exists at least one unselected standard board whose length does not exceed the residual length of the selected original board, return to step (4); otherwise, return to step (2) until each standard board node has been selected once.

Figure 5 shows the model of the solution constructed according to the above steps, and Figure 6 shows the corresponding layout scheme, assuming that there are twenty original boards and eight standard boards.

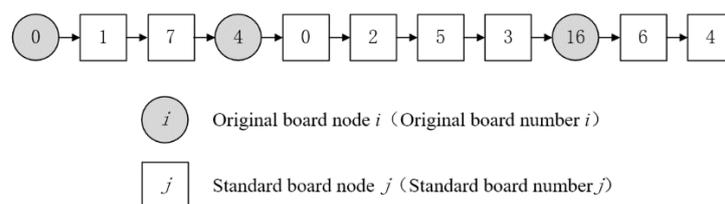


Figure 5. Solution model.

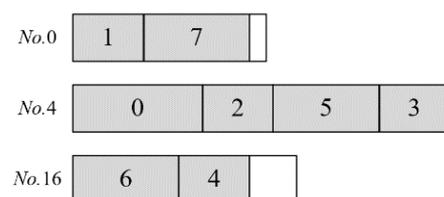


Figure 6. Layout scheme.

Figure 7 shows the encoding method; (4,0,4,4,16,4,16,0) is a encoded chromosome, with each individual in the population having only one chromosome. The length of

the chromosome is the same as the total quantity of standard boards; a chromosome corresponds to a layout scheme, with each gene of the chromosome representing an original board number.

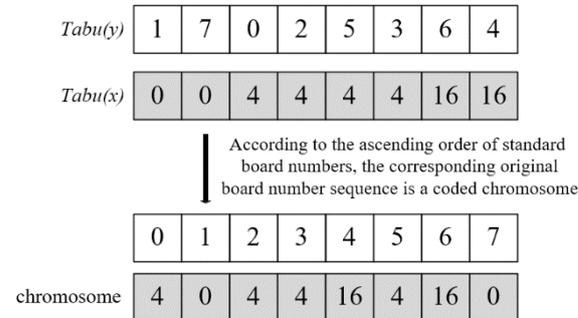


Figure 7. Encoding method.

3.3.2. Pheromone Update

In the process of constructing the solution, after the ant transfers from the original board node i to the standard board node j , the corresponding pheromone τ_{ij} should volatilize in a certain proportion, so as to reduce the selection probability of the nodes that have been selected before and enhance the ability to explore new layout schemes. The local pheromone is updated as follows:

$$\tau_{ij}(t + 1) = (1 - \chi)\tau_{ij}(t) \tag{13}$$

where χ is the local pheromone volatilization coefficient.

In order to better maintain the dynamic balance between exploration and orientation, a hybrid strategy is adopted to update the global pheromone; that is, the global optimal solution is used once every r iterations, and the iterative optimal solution is used in other iterations. The global pheromone is updated as follows:

$$\tau_{ij}(t + 1) = (1 - \rho) \tau_{ij}(t) + \rho \Delta\tau_{ij}(t) \tag{14}$$

$$\Delta\tau_{ij}(t) = \begin{cases} Q / (1 - f_{gbest}) & \text{condition 1} \\ Q / (1 - f_{dbest}) & \text{condition 2} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

condition 1 : t is an integral multiple of r and $(i, j) \in$ the global optimal solution
condition 2 : t is not an integral multiple of r and $(i, j) \in$ the iterative optimal solution

where ρ is the global pheromone volatilization coefficient, $\Delta\tau_{ij}(t)$ is the pheromone increment for the t -th iteration, Q is the pheromone enhancement factor, f_{gbest} is the fitness of the global optimal solution, and f_{dbest} is the fitness of the iterative optimal solution.

In order to avoid stagnation, the pheromone is limited between $[\tau_{min}, \tau_{max}]$, and the upper and lower bounds of the pheromone are determined by the calculation method in the literature [41]. In addition, the initial pheromone is set at $\tau_{max}(0)$ to improve the global search ability of ants in the initial period.

3.3.3. Adaptive Strategy

When the global optimal solutions of n_s consecutive generations are the same, the algorithm is considered to fall into local convergence, and the following adaptive strategies need to be adopted:

- (1) Referring to the pheromone smoothing mechanism [42], establish the weighted average between each node's pheromone and the upper bound of the pheromone to prevent premature convergence.

- (2) In order to expand the exploration range of ants, adjust the global pheromone volatilization coefficient by $\rho(t+1) = \max(0.95\rho(t), \rho_{min})$.
- (3) The value of q_0 determines the relative importance between using prior knowledge and exploring new solutions; in order to increase the probability that ants accept random solutions, adjust q_0 by $q_0 = q_0 \cdot \exp(\omega(n_s/ANT_ITER))$, where ρ_{min} is the lower bound of the global pheromone volatilization coefficient, ω is a negative number, n_s is the number of stagnations, and ANT_ITER is the maximum number of iterations of the IACA.

3.4. Immune Genetic Algorithm (IGA)

3.4.1. Fitness Function

Fitness function is a performance index to evaluate the individual quality, and the individual performance increases with fitness. Since the optimization goal of the solid wood board CSP is to maximize the board utilization rate, the objective function is taken as the fitness function:

$$f = l/L_c \quad (16)$$

where l is the total length of all standard boards, L_c is the total length of all the original boards selected in a layout scheme, and f is the fitness, namely the board utilization rate.

3.4.2. Genetic Operations

(1) Selection Operation

Roulette wheel selection is a frequently used method in GAs [43]. The selection probability of the individual is directly proportional to its fitness; this may lead to excessive concentration of similar individuals with high fitness in the population after the selection operation, to reduce the search space and increase the probability of falling into the local optimum. In order to maintain population diversity and preserve excellent individuals, the concentration regulation mechanism of the immune system is introduced. The individual selection probability is jointly determined by fitness and concentration, which can promote individuals with high fitness and inhibit individuals with high concentration. The selection probability of individual i is calculated as follows:

$$P_i = \frac{f_i \exp(\theta C_i)}{\sum_{i=1}^N (f_i \exp(\theta C_i))} \quad (17)$$

where f_i is the fitness of individual i , C_i is the concentration of individual i , N is the population size, and θ is a negative number.

Some individuals with high fitness may be inhibited because of their high concentration, resulting in the loss of these individuals. Therefore, the elite preservation strategy is adopted: ensure the individual with the highest fitness in the current population is preserved in the next generation population, and delete the individual with the lowest fitness in the next generation population.

(2) Crossover Operation

As the core operation of the GA, crossover determines the global search ability of the algorithm. In this paper, two-point crossover operation is adopted, as shown in Figure 8, and its design includes two main aspects: the crossover probability and the position of the crossover point.

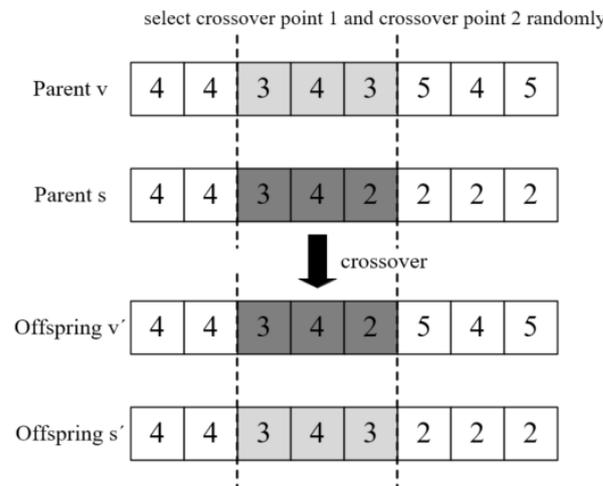


Figure 8. Two-point crossover operation.

The crossover probability directly affects the convergence of the algorithm. In this paper, adaptive crossover probability is proposed: in order to preserve excellent genes, the crossover probability decreases gradually with an increase in fitness for the individual whose fitness is higher than the average fitness of the population, and the maximum crossover probability is adopted for the individual whose fitness is lower than the average fitness of the population to generate new individuals. The crossover probability is calculated as follows:

$$P_c = \begin{cases} P_{cmin} + \frac{P_{cmax} - P_{cmin}}{1 + \exp\left(\varphi \left(\frac{2(f' - f_{avg})}{f_{max} - f_{avg}} - 1\right)\right)} & f' \geq f_{avg} \\ P_{cmax} & f' < f_{avg} \end{cases} \quad (18)$$

where P_{cmax} is the upper bound of the crossover probability, P_{cmin} is the lower bound of the crossover probability, f' is the larger fitness of the two individuals performing the crossover operation, f_{avg} is the average fitness of the population, f_{max} is the maximum fitness of the population, and φ is a constant.

The position of the crossover point determines the effectiveness of the crossover operation. If the position of the crossover point is not selected properly, the offspring after the crossover operation may be the same as the parent, which means the crossover operation is invalid. To avoid ineffective crossover, it is necessary to ensure that at least one of the two different crossover points selected randomly exists in the effective region.

Assuming that two individuals performing the crossover operation are v and s ,

$$v = [v_0, v_1, \dots, v_{B-1}], s = [s_0, s_1, \dots, s_{B-1}]$$

$$V = \{i | s_i \neq v_i, i = 0, 1, \dots, B - 1\}$$

the effective region is $[\min(V), \max(V)]$.

(3) Mutation Operation

Mutation operation can provide new genes for individual evolution and is an important means to maintain population diversity. In this paper, single-point mutation operation is adopted; it is necessary to ensure that the replaced original board number is different from the former one.

After genetic operations, all individuals in the population are checked, and the individuals that do not meet the constraints are deleted.

3.4.3. Filter Similar Individuals

When the algorithm finds a region with an extreme value, individuals are constantly concentrated in this region, which will lead to the emergence of many new similar individ-

uals. Thus, the population diversity decreases rapidly, compromising the ability to explore other new regions with extreme values and worsening the convergence efficiency. To avoid this, similar individuals need to be filtered. By referring to the self-generated diversity and maintenance mechanism of the immune system [44,45], the filter operation is performed according to the similarity threshold Th , which is beneficial to expand each individual's independent search space and reduce the singleness of population genes. The specific steps of filtering similar individuals are as follows:

- (1) For any two identical individuals in the population, delete one randomly.
- (2) Select two individuals in the population and calculate their similarity. If the similarity is not less than the similarity threshold Th , add these two individuals into region D until any two individuals in the population have been selected.
- (3) Select two individuals in region D and compare their respective concentrations. If the concentrations of these two individuals are not equal, delete the individual with higher concentration; otherwise, move to step (4).
- (4) Compare the respective fitness of the individuals mentioned in step (3). If the fitness of these two individuals is not equal, delete the individual with lower fitness; otherwise, delete one randomly.
- (5) Repeat step (3) and step (4) until any two individuals in region D have been compared.

3.4.4. Dynamic Supplementation of New Individuals

After deleting the individuals that do not meet the constraints and filtering similar individuals, in order to achieve the preset population size N , it is necessary to supplement new individuals. The method specified in Section 3.3.1 is used to generate new individuals. At the same time, if the similarity between the newly generated individual and any one individual in the population is less than the similarity threshold Th , the newly generated individual is added into the population; otherwise, it will not be preserved, which can better suppress the repetition in the search process and strengthen the global optimization ability.

4. Experiments and Discussion

In order to verify the effectiveness and superiority of the AC-IGA, experiments on the ACA [46], GA [47], GWO [48], PBO [49] and AC-IGA were carried out by PyCharm, and their experimental results are compared. The computer operating system is Windows10, the processor is Inter (R) Xeon (R) W-2155 CPU@3.30 GHz, and the RAM is 64 GB. Table 1 shows the parameter values of the AC-IGA. To ensure the effectiveness of the comparison, the values of the same parameters in the five algorithms are the same.

Table 1. Parameter values.

Parameter	ω	ANT_ITER	ρ_{min}	Th	q_0	r	Q	n_s	α	β
Value	−0.5	50	0.1	0.8	0.6	5	1	5	2	4.5
Parameter	θ	IGA_ITER	P_{cmin}	P_{cmax}	ϕ	P_m	N	χ	ρ	
Value	−0.8	50	0.6	0.9	9.903438	0.1	60	0.1	0.5	

At present, there are no standard instances to test the one-dimensional cutting stock problem. Therefore, according to the real data provided by a furniture enterprise, we generated 12 instances for each of the nine classes with the random number generator [50], resulting in a total of 108 instances. The length range of original boards is [3000, 12000], and the length range of standard boards is [300, 4500]. Each generated instance is solved by ACA, GA, GWO, PBO and AC-IGA five consecutive times. The parameters for generating instances and the experimental results are presented in Table 2, where n is the number of types of original boards, m is the number of types of standard boards, d is the total number of standard boards (its value represents the range of the data generated randomly 12 times), u is the average of the experimental results under the corresponding class (average utilization rate), and Arg is the cumulative average of each column of data (the

average of the experimental results of 108 instances). Figure 9 shows the comparison diagram of the statistical experimental results.

Table 2. Parameters and experimental results.

Class	n	m	d	AC-IGA	ACA	GA	GWO	PBO
				u (%)				
1	1	4	[20, 50)	83.230	82.335	81.818	82.618	81.276
2	2	5	[50, 100)	95.984	90.647	90.468	90.449	87.853
3	3	5	[100, 150)	98.574	96.316	96.003	96.199	94.360
4	3	6	[150, 200)	98.566	95.713	95.454	95.456	93.729
5	4	7	[200, 250)	98.932	95.909	95.860	95.851	94.727
6	5	8	[250, 300)	99.212	96.695	96.631	96.727	95.695
7	6	9	[300, 350)	98.895	97.799	97.719	97.774	96.967
8	8	10	[350, 400)	98.994	97.769	97.757	97.797	97.096
9	10	15	[400, 450)	99.060	97.501	97.409	97.458	96.727
<i>Arg</i> (%)				96.828	94.520	94.346	94.481	93.159

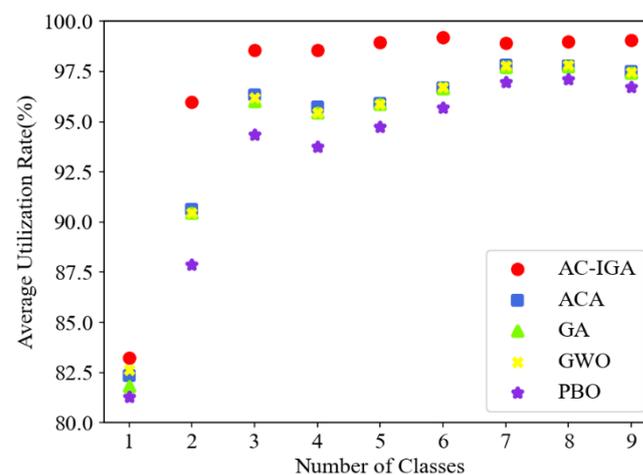


Figure 9. Comparison diagram of the statistical experimental results.

As can be seen from Table 2 and Figure 9, the average utilization rate of each class obtained by the AC-IGA is better than the other four algorithms. In addition, the cumulative average utilization rate of 108 instances obtained by the AC-IGA is 2.308% higher than that of the ACA, 2.482% higher than that of the GA, 2.347% higher than that of the GWO, and 3.669% higher than that of the PBO. It can be seen that when the solutions of the four comparison algorithms have high precision, the average utilization rate obtained by the AC-IGA still shows further improvement, which verifies the effectiveness of the AC-IGA in improving the solution quality and its superiority in solving the one-dimensional cutting stock problem.

In order to further verify the application value of the AC-IGA in solving the solid wood board cutting stock problem, a production instance of a furniture enterprise was selected for layout calculation. The production instance was solved by the ACA, GA, GWO, PBO and AC-IGA ten consecutive times, and each algorithm obtained ten results. Table 3 shows the data for the production instance.

Table 3. Data for the production instance.

Original board	Length/mm	9400	8200	7550	5500			
	Supply	22	26	28	38			
Standard board	Length/mm	1390	980	830	785	660	580	450
	Demand	40	35	35	35	35	35	35

Table 4 shows the analysis of the experimental results, where Min, Max, and Mean are the minimum, maximum, and average of the experimental results obtained by the corresponding algorithm, and SD is the standard deviation. Compared with the ACA, GA, GWO, and PBO, the Min obtained by the AC-IGA increased by 3.157%, 3.290%, 3.090%, and 4.801%, respectively, the Max obtained by the AC-IGA increased by 2.974%, 2.771%, 2.839%, and 3.846%, respectively, the Mean obtained by the AC-IGA increased by 2.974%, 3.034%, 2.949%, and 4.063%, respectively, and the SD of the AC-IGA is the smallest, which indicates that the AC-IGA has better optimization ability and robustness.

Table 4. Analysis of the experimental results.

Algorithm	Utilization Rate of Board (%)			
	Min	Max	Mean	SD
AC-IGA	98.929	99.240	99.051	0.097
ACA	95.772	96.266	96.077	0.149
GA	95.639	96.469	96.017	0.244
GWO	95.839	96.401	96.102	0.154
PBO	94.128	95.394	94.988	0.379

Figure 10 shows the relationship between the highest board utilization rate searched by the five algorithms and the number of iterations; the red solid line is AC-IGA, the blue dashed line is ACA, the green dash-dot line is GA, the yellow solid line is GWO, and the purple dotted line is PBO.

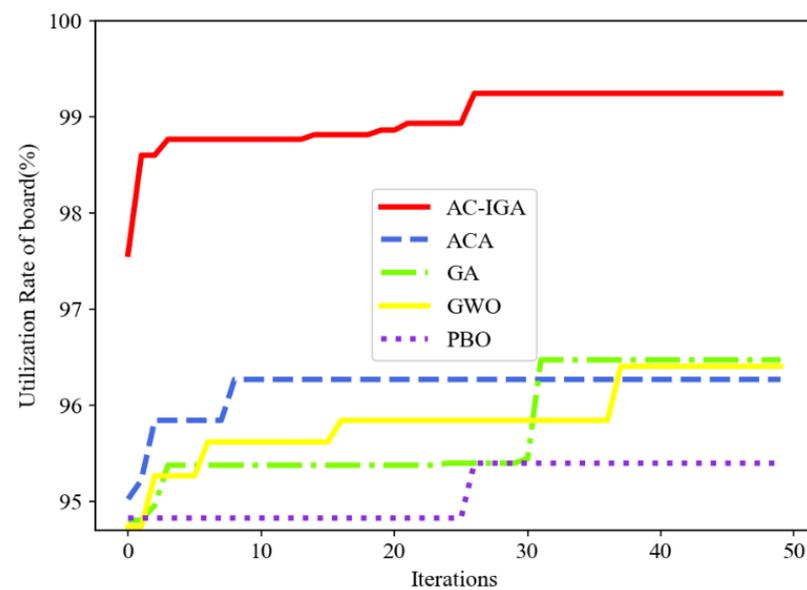


Figure 10. Iterative convergence curves.

The board utilization rate of the AC-IGA converges to 99.240% after twenty-seven iterations and the board utilization rate of the GA converges to 96.469% after thirty-two iterations, which shows that the AC-IGA can find the optimal solution faster, and the

solution quality is better, than the GA. This is mainly because the performance of the GA is excessively dependent on the initial population’s quality; the quality of the initial solutions generated randomly is low, which results in low solving efficiency. In this paper, the solutions generated by each generation of IACA are used as the initial population of AC-IGA, which can improve the solution quality and increase the convergence speed.

The board utilization rate of the ACA converges to 96.266% after nine iterations, which shows that although the AC-IGA is slower than the ACA in finding the optimal solution, the solution quality of the AC-IGA is better. This is mainly because ants are concentrated on some local solutions earlier under the action of the positive feedback mechanism of the ACA, which reduces the population diversity and makes it difficult for ants to further explore new solutions. In this paper, the IGA is introduced into each iteration of the IACA, and the solutions with higher board utilization rate can be obtained through multiple iterations of the IGA. Improved methods (including pheromone update, adaptive strategy, and filter similar individuals) are adopted, which can effectively maintain the population diversity in the evolutionary process, allows for the constant exploration of new layout schemes with higher board utilization rates, and improves the ability to jump out of the local optimum.

In addition, the board utilization rate of the GWO converges to 96.401% after thirty-eight iterations and the board utilization rate of the PBO converges to 95.394% after twenty-seven iterations, which shows that the quality of the solutions found by the AC-IGA is better than that found by the GWO and PBO. This is mainly because the GWO and PBO generated initial populations by random method, which cannot ensure good population diversity. In the GWO, individuals update their location information only based on three optimal individuals (α , β , and δ). Individuals are independent of each other and lack of effective information exchange, resulting in slow convergence speed. If α is the local optimal solution, the GWO will converge prematurely. Besides, the convergence factor used in the GWO decreases linearly with the number of iterations, which does not conform to the actual search process of the GWO and cannot balance the global exploration ability and local development ability.

In summary, the AC-IGA achieved great improvement in solution precision and convergence, and thus is an effective method to solve the solid wood board CSP.

Table 5 shows the optimal layout scheme obtained by the AC-IGA, where I is the layout mode number, II is the length of the original board, III is the length (quantity) of standard boards, IV is the residual length of the original board, and V is the quantity of the layout mode. A total of thirty-one original boards are selected in this layout scheme, and the board utilization rate is 99.240%.

Table 5. The optimal layout scheme.

I	II	III	IV	V	I	II	III	IV	V
1	5500	1390 (3); 830; 450	50	4	15	7550	1390 (4); 980; 830	180	1
2	5500	1390 (3); 660; 580	90	1	16	7550	1390 (2); 980; 830; 785 (3); 580	25	1
3	5500	1390 (2); 980; 830 (2)	80	1	17	7550	980 (6); 830 (2)	10	1
4	5500	1390; 980; 830 (2); 660 (2)	150	1	18	7550	660 (3); 580 (8); 450 (2)	30	1
5	5500	1390 (3); 660 (2)	10	1	19	7550	660 (6); 580 (3); 450 (4)	50	1
6	5500	980 (5); 580	20	1	20	8200	1390 (2); 980 (2); 830; 785 (2);	30	1
7	5500	980 (4); 785 (2)	10	1			580; 450		

Table 5. *Cont.*

I	II	III	IV	V	I	II	III	IV	V
8	5500	980 (2); 830 (3); 580; 450	20	1	21	8200	1390 (2); 980 (4); 830; 660	10	1
9	5500	830 (5); 660 (2)	30	1	22	8200	830 (6); 785 (2); 580 (2); 450	40	1
10	5500	830 (4); 785; 660 (2)	75	1	23	8200	785 (8); 660 (2); 580	20	1
11	5500	785 (4); 660 (2); 580; 450	10	2	24	9400	1390 (4); 980; 830; 785 (2); 450	10	1
12	5500	785 (5); 580; 450 (2)	95	1	25	9400	1390 (5); 980 (2); 450	40	1
13	5500	660 (5); 580(2); 450 (2)	140	1	26	9400	980 (5); 830 (2); 785 (2); 580 (2)	110	1
14	5500	660 (4); 580; 450 (5)	30	1	27	9400	660; 580 (8); 450 (9)	50	1

5. Conclusions

This paper investigated the solid wood board cutting stock problem and established a mathematical model, with the optimization goal of the highest possible board utilization rate. An ant colony-immune genetic algorithm (AC-IGA) was proposed to solve this model. The advantages of the AC-IGA are as follows:

(1) To avoid the problem of premature convergence in the ant colony algorithm, we improved the pheromone update method. The adaptive strategy was adopted to jump out of the local optimum when the AC-IGA stagnates for an extended time. In addition, the concentration mechanism and genetic mechanism of individual diversity of the immune system was introduced into the GA, which effectively maintained the population diversity and enhanced the global search ability.

(2) By introducing the IGA into each iteration of the IACA, the global search ability and local exploitation ability could achieve better balance, and the solution quality and search efficiency was greatly improved.

Compared with the ACA, GA, GWO, and PBO, the experimental results showed that the AC-IGA can obtain a higher board utilization rate on the premise of ensuring robustness. This proves that the AC-IGA is a good approach to solve the solid wood board cutting stock problem and is more conducive to maximizing profits for furniture enterprises.

Author Contributions: Conceptualization, M.T. and Y.L.; methodology, M.T.; validation, M.T.; writing—original draft preparation, M.T.; writing—review and editing, Y.L., F.D. and Z.W.; project administration, Y.L.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the 2019 Jiangsu Province Key Research and Development Plan by the Jiangsu Province Science and Technology under grant BE2019112.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, M.Y.; Zhang, Z.F. Application of Material Characteristics in Furniture Design. *Packag. Eng.* **2017**, *38*, 141–145.

2. Tankut, N.; Tankut, A.N.; Zor, M. Finite Element Analysis of Wood Materials. *Wood Ind. Drv. Ind.* **2014**, *65*, 159–171. [[CrossRef](#)]
3. Zhen, W.; Wang, A.; Zhuo, M. Research of Using Laser Direct Marking Two-Dimensional Car Code Technology on Wood. In Proceedings of the 2015 6th International Conference on Manufacturing Science and Engineering, Guangzhou, China, 28–29 November 2015; Volume 32.
4. PRC, S. The Sixth National Forest Resources Inventory and the Status of Forest Resources. *Green China* **2005**, *2*, 10–12.
5. Jing-Chu, H.U. Discussion on Strategy about Solid Wood Furniture Sustainable Development. *China For. Prod. Ind.* **2003**, *5*, 9–13. [[CrossRef](#)]
6. Liu, L.; Sun, H.G. The Impact of Collective Forestland Tenure Reform on the Forest Economic Efficiency of Farmers in Zhejiang Province. *Sustainability* **2019**, *11*, 2272. [[CrossRef](#)]
7. Wang, L.; Huang, Z.; Gao, H.D.; Ni, C. A method for continuously monitoring the quality of Masson pine seedlings. *PLoS ONE* **2021**, *16*, e0234592. [[CrossRef](#)]
8. Sun, X.; Katsigris, E.; White, A. Meeting China's Demand for Forest Products: An Overview of Import Trends, Ports of Entry, and Supplying Countries, with Emphasis on the Asia-Pacific Region. *Int. For. Rev.* **2004**, *6*, 227–236. [[CrossRef](#)]
9. Wargula, L.; Kukla, M. Determination of Maximum Torque during Carpentry Waste Comminution. *Wood Res.* **2020**, *65*, 771–783. [[CrossRef](#)]
10. Ihnat, V.; Lubke, H.; Balbercak, J.; Kuna, V. Size Reduction Downcycling of Waste Wood. Review. *Wood Res.* **2020**, *65*, 205–220. [[CrossRef](#)]
11. Gordic, D.; Babic, M.; Jelic, D.; Koncalovic, D.; Vukasinovic, V. Integrating Energy and Environmental Management in Wood Furniture Industry. *Sci. World J.* **2014**, *2014*, 1–18. [[CrossRef](#)]
12. Vanzela, M.; Melega, G.M.; Rangel, S.; de Araujo, S.A. The integrated lot sizing and cutting stock problem with saw cycle constraints applied to furniture production. *Comput. Oper. Res.* **2017**, *79*, 148–160. [[CrossRef](#)]
13. Xiong, X.Q.; Guo, W.J.; Fang, L.; Zhang, M.; Wu, Z.H.; Lu, R.; Miyakoshi, T. Current state and development trend of Chinese furniture industry. *J. Wood Sci.* **2017**, *63*, 433–444. [[CrossRef](#)]
14. Guo, Y.Q.; Li, Z.Y.; Yang, X.L.; Li, J.B. Research on Shaking Table Test of Earthquake Simulation Based on Hybrid Integration Algorithm. *IEEE Access* **2020**, *8*, 208961–208968. [[CrossRef](#)]
15. Savsani, V.; Savsani, P.; Arya, P. Effect of Applying Advanced Optimization Techniques for the One-Dimensional Cutting Stock Problem. In Proceedings of the ASME International Mechanical Engineering Congress and Exposition, Montreal, QC, Canada, 14–20 November 2014; Volume 11.
16. Shi, Y.Y.; Li, Q.Q.; Bu, S.Q.; Yang, J.F.; Zhu, L.F. Research on Intelligent Vehicle Path Planning Based on Rapidly-Exploring Random Tree. *Math. Probl. Eng.* **2020**, *2020*, 1–14. [[CrossRef](#)]
17. Fayyaz, S.; Sattar, M.K.; Waseem, M.; Ashraf, M.U.; Alsubhi, K. Solution of Combined Economic Emission Dispatch Problem Using Improved and Chaotic Population Based Polar Bear Optimization Algorithm. *IEEE Access* **2021**, *9*, 56152–56167. [[CrossRef](#)]
18. Pleszczyński, M.; Zielonka, A.; Połap, D.; Woźniak, M.; Mańdziuk, J. Polar Bear Optimization for Industrial Computed Tomography with Incomplete Data. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 28 June–1 July 2021.
19. Saddique, M.W.; Haroon, S.S.; Amin, S.; Bhatti, A.R.; Liaqat, R. Optimal Placement and Sizing of Shunt Capacitors in Radial Distribution System Using Polar Bear Optimization Algorithm. *Arab. J. Sci. Eng.* **2021**, *46*, 873–899. [[CrossRef](#)]
20. Sikora, A.; Zielonka, A.; Woźniak, M. Heuristic Optimization of 18-Pulse Rectifier System. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 28 June–1 July 2021.
21. Jiang, T. Low-carbon workshop scheduling problem based on grey wolf optimization. *Comput. Integr. Manuf. Syst.* **2018**, *24*, 2428–2435.
22. Liu, E.; Yao, X.; Liu, M.; Jin, H. AGV path planning based on improved grey wolf optimization algorithm and its implementation prototype platform. *Comput. Integr. Manuf. Syst.* **2018**, *24*, 2779–2791.
23. Mostafa, A.; Salah, K.; Mohamed, E.; Yu, J.; Francisco, J. Solving Non-Smooth Optimal Power Flow Problems Using a Developed Grey Wolf Optimizer. *Energies* **2018**, *11*, 1692.
24. Durasević, M.; Jakobović, D. Evolving dispatching rules for optimising many-objective criteria in the unrelated machines environment. *Genet. Program. Evolvable Mach.* **2018**, *19*, 9–51. [[CrossRef](#)]
25. Sun, P.; Wu, J.; Liao, M.; Zhang, J. Battlefield resource dynamic scheduling model and algorithm based on improved self-adaptive genetic algorithm. *Syst. Eng. Electron.* **2018**, *40*, 2459–2465.
26. Turk, A.; Sara, D.; Ozkok, M.; Kose, E. Minimization of total trim loss occurring in pipe cutting in shipbuilding with genetic algorithm. *J. Eng. Marit. Environ.* **2021**, *235*, 695–704. [[CrossRef](#)]
27. Sarper, H.; Jaksic, N.I. Evaluation of procurement scenarios in one-dimensional cutting stock problem with a random demand mix. *Procedia Manuf.* **2018**, *17*, 827–834. [[CrossRef](#)]
28. Cui, Y.D.; Song, X.; Chen, Y.; Cui, Y.P. New model and heuristic solution approach for one-dimensional cutting stock problem with usable leftovers. *J. Oper. Res. Soc.* **2017**, *68*, 269–280. [[CrossRef](#)]
29. Cerqueira, G.R.L.; Aguiar, S.S.; Marques, M. Modified Greedy Heuristic for the one-dimensional cutting stock problem. *J. Comb. Optim.* **2021**, 1–18. [[CrossRef](#)]
30. Campello, B.S.C.; Ghidini, C.T.L.S.; Ayres, A.O.C.; Oliveira, W.A. A residual recombination heuristic for one-dimensional cutting stock problems. *Top* **2021**, 1–27. [[CrossRef](#)]

31. Ben Lagha, G.; Dahmani, N.; Krichen, S. Particle Swarm Optimization Approach for Resolving the Cutting Stock Problem. In Proceedings of the 2014 International Conference on Advanced Logistics & Transport (ICALT 2014), Hammamet, Tunisia, 1–3 May 2014; pp. 259–263.
32. Lu, Q.; Zhou, X. GPU parallel ant colony algorithm for the dynamic one-dimensional cutting stock problem based on the on-line detection. *Chin. J. Entific. Instrum.* **2015**, *36*, 1774–1782.
33. Hua, X.; Zhang, Q.; Fu, W.; Zhang, X.; Li, J. A global heuristic and two-tier algorithm to solve power cable optimal cutting problem. *Power Syst. Technol.* **2012**, *36*, 6.
34. Benjaoran, V.; Bhokha, S. Three-step solutions for cutting stock problem of construction steel bars. *KSCE J. Civ. Eng.* **2014**, *18*, 1239–1247. [[CrossRef](#)]
35. Pitombeira-Neto, A.R.; Prata, B.D. A matheuristic algorithm for the one-dimensional cutting stock and scheduling problem with heterogeneous orders. *Top* **2020**, *28*, 178–192. [[CrossRef](#)]
36. Bai, X.; Runjia, L.I.; Jingtao, X.U.; Song, E. Wood surface defect image segmentation based on Graph Cut algorithm. *J. For. Eng.* **2018**, *3*, 116–122.
37. Dai, X.; Xu, Y.; Chen, J.; Zheng, J. Evaluation of Mixing Uniformity for Inline Mixers by Image Processing. *Trans. ASABE* **2020**, *63*, 429–443. [[CrossRef](#)]
38. Huang, L.; Xie, F.; Shen, S.; Zhao, J.; Lu, R. Human Emotion Recognition Based on Face and Facial Expression Detection Using Deep Belief Network Under Complicated Backgrounds. *Int. J. Pattern Recognit. Artif. Intell.* **2020**, *34*, 2056010. [[CrossRef](#)]
39. Dou, S.H.; Liu, G.Y.; Yang, Y.B. A New Hybrid Algorithm for Cold Chain Logistics Distribution Center Location Problem. *IEEE Access* **2020**, *8*, 88769–88776. [[CrossRef](#)]
40. Zheng, Y.; Luo, Q.; Wang, H.B.; Wang, C.H.; Chen, X. Path planning of mobile robot based on adaptive ant colony algorithm. *J. Intell. Fuzzy Syst.* **2020**, *39*, 5329–5338. [[CrossRef](#)]
41. Pan, H.; You, X.; Liu, S. High-Frequency Path Mining-Based Reward and Punishment Mechanism for Multi-Colony Ant Colony Optimization. *IEEE Access* **2020**, *8*, 155459–155476. [[CrossRef](#)]
42. Liu, A.Z.; Deng, G.S.; Shan, S.M. Mean-contribution ant system: An improved version of ant colony optimization for traveling salesman problem. *Lect. Notes Comput. Sci.* **2006**, *4247*, 489–496.
43. Guo, X.C.; Su, J.; Zhou, H.; Liu, C.; Cao, J.; Li, L. Community Detection Based on Genetic Algorithm Using Local Structural Similarity. *IEEE Access* **2019**, *7*, 134583–134600. [[CrossRef](#)]
44. Asadujjaman, M.; Rahman, H.F.; Chakraborty, R.K.; Ryan, M. An Immune Genetic Algorithm for Solving NPV-based Resource Constrained Project Scheduling Problem. *IEEE Access* **2021**, *9*, 26177–26195. [[CrossRef](#)]
45. Corus, D.; Oliveto, P.S.; Yazdani, D. Artificial immune systems can find arbitrarily good approximations for the NP-hard number partitioning problem. *Artif. Intell.* **2019**, *274*, 180–196. [[CrossRef](#)]
46. Wu, Z.; Zhang, L.; Wang, K. An Ant Colony Algorithm for One-Dimensional Cutting-Stock Problem. *Mech. Sci. Technol. Aerosp. Eng.* **2008**, *27*, 1681–1684.
47. Liu, Z.; Weng, X.; Jing, W.; Xia, X. Optimum cutting of multi-size stock tubing or profile based on genetic algorithm. *Comput. Era* **2018**, *12*, 67–70, 74.
48. Zhou, C.; Yuan, Z.; Lei, J.; Bai, H.; Tan, Y.; Tao, S. Optimal Planning of Integration of Decentralized Wind Power into Distribution Network Based on Grey Wolf Optimizer Algorithm. *Proceedings* **2020**, *32*, 73–80. [[CrossRef](#)]
49. Dawid, P.; Marcin, W. Polar Bear Optimization Algorithm: Meta-Heuristic with Fast Population Movement and Dynamic Birth and Death Mechanism. *Symmetry* **2017**, *9*, 203.
50. Poldi, K.C.; Arenales, M.N. Heuristics for the one-dimensional cutting stock problem with limited multiple stock lengths. *Comput. Oper. Res.* **2009**, *36*, 2074–2081. [[CrossRef](#)]