

Article

# Process Planning for Large Container Ship Propeller Shaft Machining Based on an Improved Ant Colony Algorithm

Guotai Du <sup>1</sup>, Hongkui Ma <sup>2</sup>, Yu Bai <sup>3</sup> and Ning Mei <sup>1,\*</sup>

<sup>1</sup> College of Engineering, Ocean University of China, Qingdao 266005, China; 13012785474@163.com

<sup>2</sup> Qingdao Hiron Commercial Cold Chain Co., Ltd., Qingdao 266400, China; mhk@chinahiron.com

<sup>3</sup> College of Mechanical and Electrical Engineering, Qingdao City University, Qingdao 266106, China; yu.bai@qdc.edu.cn

\* Correspondence: nmei@ouc.edu.cn

**Abstract:** To accommodate the production and manufacture of complex and customized marine components and to avoid the empirical nature of process planning, machining operations can be automatically sequenced and optimized using ant colony algorithms. However, traditional ant colony algorithms exhibit issues in the context of machining process planning. In this study, an improved ant colony algorithm is proposed to address these challenges. The introduction of a tiered distribution of initial pheromones mitigates the blindness of initial searches. By incorporating the number of iterations into the expectation heuristic function and introducing a ‘reward–penalty system’ for pheromones, the contradictions between convergence speed and the tendency to fall into local optima are avoided. Applying the improved ant colony algorithm to the process planning of large container ship propeller shaft machining, this study constructs a ‘distance’ model for each machining unit and develops a process constraint table. The results show significant improvements in initial search capabilities and convergence speed with the improved ant colony algorithm while also resolving the contradiction between convergence speed and optimal solutions. This verifies the feasibility and effectiveness of the improved ant colony algorithm in intelligent process planning for ships.

**Keywords:** ant colony algorithm; process planning; large container ship propeller shaft; ship intelligent manufacturing



**Citation:** Du, G.; Ma, H.; Bai, Y.; Mei, N. Process Planning for Large Container Ship Propeller Shaft Machining Based on an Improved Ant Colony Algorithm. *J. Mar. Sci. Eng.* **2024**, *12*, 841. <https://doi.org/10.3390/jmse12050841>

Academic Editor: Sergei Chernyi

Received: 22 April 2024

Revised: 9 May 2024

Accepted: 13 May 2024

Published: 18 May 2024



**Copyright:** © 2024 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 advent of Industry 4.0 and Industry 5.0, the shipbuilding industry has gradually transitioned towards intelligence and digitization [1]. However, the complexity of multivariable and time-varying factors remains a major challenge in the production of ship components [2]. Specifically, the production of large container ship propeller shafts, characterized by complex processes and customization, renders traditional process planning methods inadequate. Consequently, the role of computer-aided process planning as a bridge between customized design and actual manufacturing is becoming increasingly prominent. Statistics show that computer-aided process planning can reduce the workload of process planning for new parts or products by approximately 58% [3]. Traditional process planning methods, being overly reliant on experience, often lead to the generation of multiple process routes for a single propeller shaft, necessitating numerous trial-and-error experiments for time and cost estimation, resulting in inefficiency in the production process [4]. Currently, there is a global effort to find effective solutions to these challenges and to explore how to more quickly and effectively complete process planning for different components [5]. Therefore, the quick and efficient automation of process planning in the context of complex, multi-process, and customized manufacturing has become an urgent problem to solve.

Process planning in machining involves selecting and sequencing machining operations (such as milling, drilling, and turning) for a given part, in compliance with design

standards and manufacturing practices. It is one of the most complex dynamic decision-making problems and a key aspect of computer-aided process planning. Researchers have proposed various methods for process planning, including process reuse methods, rule-based inference methods, machine learning-based methods, and heuristic algorithm optimization-based methods, each with its own advantages and disadvantages, as depicted in Table 1. Over the past few decades, heuristic algorithm optimization-based methods have become increasingly mature and stable. Compared to other heuristic algorithms, the ant colony algorithm stands out for its few adjustable parameters, strong positive feedback capability, and ability to adapt to changes and handle complex dynamic environments. Consequently, it has been widely applied to path planning problems in domains such as robotics, network routing, and logistics. Zhang [6] utilized an improved elite ant colony optimization algorithm to solve the grain emergency vehicle dispatch model, demonstrating its effectiveness in distribution scheduling. Sara Perez-Carabaza et al. [7] used an ant colony algorithm to enable UAV trajectory optimization planners to quickly obtain high-quality solutions. Xiang et al. [8] proposed an ant colony algorithm based on the diversity of demand coverage, maintaining diversity in routes to effectively respond to emerging customer requests. The increasing complexity of customized large parts has extended production cycles when relying solely on experiential process route planning. Traditional ant colony algorithms suffer from issues like prolonged reaction times, significant initial search blindness, and the trade-off between convergence speed and local optima. These challenges often result in extended processing times, imprecise process planning routes, or even non-convergence in multi-step processing planning. Hence, this study proposes an enhanced ant colony algorithm tailored for process planning. It aims to mitigate local optima traps, enhance initial search capabilities, optimize convergence speed, and potentially offer effective solutions for process planning of complex, customized large parts. In the realm of intelligent manufacturing within the marine equipment sector, the fabrication of propeller shafts for large container ships stands out for its significant complexity and high degree of customization. In this study, focusing on the production of large container ship propeller shafts, the shaft machining process planning problem is analogized to a path optimization problem. A 'distance' model for various machining units with differences and similarities is established. Using the improved ant colony algorithm, machining units are selected and sequenced according to the constraint table, resulting in an optimal process planning route. By innovating the initial pheromone distribution method, a pheromone tiered distribution based primarily on basic process planning principles is proposed. Analyzing the optimal process routes in the first iterations of multiple tests, the improved ant colony algorithm effectively resolves the issue of weak initial search capabilities of the traditional algorithm, enhancing the guidance of subsequent searches. Furthermore, by innovating the expectation heuristic function and pheromones updating methods, a heuristic function factor based on the number of iterations and a pheromone update method based on the 'reward-penalty system' are proposed. A decay coefficient that changes with the number of iterations is incorporated into the expectation heuristic function, strengthening the role of pheromones in the later stages of iterative searches and avoiding the contradiction of increased convergence speed leading to local optima. This study aims to apply the improved ant colony algorithm to the process planning of large container ship propeller shafts, addressing issues such as empirical process routes, blindness in algorithm searches, and contradictions in optimal convergence. It provides technical and methodological support for the transition from Industry 4.0 to Industry 5.0 in intelligent manufacturing and data support for the industrial application of the improved ant colony algorithm.

**Table 1.** Comparison of various process planning methods.

Method	Researchers	Characteristics
Process reuse method	Deng et al. [9] Huang et al. [10] Kowalski et al. [11]	This method is commonly adopted by enterprises but suffers from inflexibility, leading to localized and fragmented process reasoning.
Rule-based reasoning method	Y. Zhang et al. [12] Ma et al. [13] Kang et al. [14]	This method improves the efficiency of process planning systems, but predefined rules may cause confusion due to conflicting rules.
Machine learning-based method	Sugisawa et al. [15] C. Zhang et al. [16] Zhao et al. [17]	This newly emerging method lacks interpretability and fails to consider processing efficiency.
Ant colony algorithm optimization method	Toaza et al. [18] (Gong et al. [19] Liu et al. [20]	This approach is characterized by positive feedback, high robustness, and parallelism, but its initial search capability is poor, and it tends to get trapped in local optima.
Genetic algorithm optimization method	Gómez et al. [21] Jing et al. [22] Sreenivasan et al. [23]	This approach has excellent robustness and global optimization capabilities, but it has long solution times and requires adjustment of multiple parameters.
Particle swarm algorithm optimization method	Sarvaiya et al. [24] Peng et al. [25] Petrović et al. [26]	This approach has high precision and fast convergence but often gets stuck in local optima when dealing with complex problems.

## 2. Description of the Ant Colony Algorithm

### 2.1. Traditional Ant Colony Algorithm

The ant colony algorithm (ACA) simulates the foraging paths of ants in nature to solve combinatorial path optimization problems. The more ants there are on a path, the higher the concentration of pheromones, leading to an increased likelihood of that path being chosen by other ants.

Assuming at time  $t$ , the  $m$  ant is located at point  $i$ , its transition probability  $P_{ij}^m$  is determined by Equations (1) and (2).

$$P_{ij}^m = \begin{cases} \tau_{ij}^\alpha(t) \times \eta_{ij}^\beta(t) / \sum_{s \in C} \tau_{is}^\alpha(t) \times \eta_{is}^\beta(t), & j \in C \\ 0, & j \notin C \end{cases} \quad (1)$$

$$\eta_{ij}(t) = 1/d_{ij} \quad (2)$$

where  $C$  represents the set of points that the ants allowed to select at time  $t + 1$ ;  $\tau_{ij}^\alpha(t)$  is the pheromone function on the path  $\langle i, j \rangle$  at time  $t$ ,  $\eta_{ij}^\beta(t)$  is the expectation heuristic function on the path  $\langle i, j \rangle$  at time  $t$ ;  $\alpha$  is the heuristic factor,  $\beta$  is the expectation heuristic factor, and  $d_{ij}$  is the Euclidean distance between points  $i$  and  $j$ .

To avoid excessive accumulation of pheromones, the pheromone level is updated after each iteration. The pheromone level at time  $t + 1$  is determined by Equation (3).

$$\left\{ \begin{array}{l} \tau_{ij}(t + 1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \\ \Delta\tau_{ij}(t) = \sum_{m=1}^n \Delta\tau_{ij}^m(t) \end{array} \right\} \quad (3)$$

where  $\rho$  represents the pheromone volatility coefficient, with  $0 < \rho < 1$ ,  $\Delta\tau_{ij}(t)$  denotes the total increase in pheromone level on the path  $\langle i, j \rangle$ , and  $\Delta\tau_{ij}^m(t)$  indicates the amount of pheromone deposited by the  $m$  ant on the path  $\langle i, j \rangle$  during the current iteration.

Different pheromone update methods lead to different calculations of pheromone increment. Among these, the ant cycle system, represented by Equation (4), is the most commonly used.

$$\Delta\tau_{ij}^m(t) = \begin{cases} Q/L_m, & \text{ant } m \text{ passed through path } \langle i, j \rangle . \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where  $Q$  is a constant representing the total amount of pheromone, and  $L_m$  is the total length of the path traveled by the  $m$  ant in this iteration.

## 2.2. Improved Ant Colony Algorithm

### 2.2.1. Initial Pheromones

In ACA, the initial pheromone concentration is set to the total amount of pheromone  $Q$ , which results in the initial searches being blind and random, with limited subsequent guidance, making the algorithm more challenging. Hence, this study adopts a tiered pheromone concentration distribution to avoid blind initial searches, as shown in Equation (5).

$$\tau_{ij}(0) = \{\delta * Q\}, j \in C \quad (5)$$

Here,  $\delta$  represents the weight of the initial pheromone distribution with a value between  $0 < \delta \leq 1$ , and  $Q$  is the total amount of pheromone.

### 2.2.2. Expectation Heuristic Function

From Equation (2), it can be observed that the expectation heuristic function value is inversely proportional to the distance from the next node to the destination point, encouraging ants to choose shorter distances. However, in the later stages of the search, more attention should be given to the influence of pheromone concentration on ant selection while reducing the impact of the expectation heuristic function. Therefore, in this study, a self-regulating coefficient  $\xi$  is introduced into the expectation heuristic function to reduce its impact on the transition probability in the later stages of the search, as shown in Equations (6) and (7).

$$\eta_{ij}(t) = (1 / d_{ij}) \times \xi \quad (6)$$

$$\xi = \begin{cases} I_{\max} - I / I_{\max}, & I = I_{\max} \\ 1 / I_{\max}, & I \neq I_{\max} \end{cases} \quad (7)$$

Here,  $\xi$  is the self-regulating coefficient of the expectation heuristic function,  $I$  represents the current iteration number, and  $I_{\max}$  is the maximum iteration number.

### 2.2.3. Pheromone Update Rules

In ACA, the pheromone update mechanism does not incorporate scenarios where inferior paths are encountered during the iteration. In this study, on the basis of local and global updates, a ‘reward–penalty system’ is introduced. It ‘rewards’ the best paths and ‘penalizes’ the worst paths during the current iteration, as shown in Equation (8). To prevent the algorithm from stagnating too early, upper and lower limits are set for pheromone concentration, constraining it within the range  $[\tau_{\max}, \tau_{\min}]$ . The maximum and minimum values of pheromone are calculated using Equations (9)–(11).

$$\Delta\tau_{ij} = \begin{cases} Q/L_{best}, & \text{path } \langle i, j \rangle \text{ belongs the optimal path} \\ -Q/L_{worst}, & \text{path } \langle i, j \rangle \text{ belongs the worst path} \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

Here,  $L_{best}$  represents the best path in the current iteration, and  $L_{worst}$  represents the worst path in the current iteration.

$$\tau_{ij}(t) = \begin{cases} \tau_{max}, & \tau_{ij}(t) \geq \tau_{max} \\ \tau_{ij}(t), & \tau_{min} \leq \tau_{ij}(t) \leq \tau_{max} \\ \tau_{min}, & \tau_{ij}(t) \leq \tau_{min} \end{cases} \quad (9)$$

$$\tau_{max} = Q / (1 - \rho) f_{best} \quad (10)$$

$$\tau_{min} = \tau_{max} / 2n \quad (11)$$

Here,  $f_{best}$  represents the best path in each iteration, and  $n$  is the number of nodes traversed by the ants.

### 3. Application of the Improved Ant Colony Algorithm in Process Planning

#### 3.1. Construction of a Process Planning Model

The production and manufacturing of large container ship propeller shafts belong to a personalized customization process. Traditional process planning relies heavily on the experience of workers, resulting in low efficiency and accuracy. The goal of process planning is to obtain the optimal mechanical machining sequence combination while meeting processing requirements. By comparing the differences and similarities in the machinery, tools, fixtures, clamping positions, and processing characteristic units used in each machining unit of the large container ship propeller shaft, a 'distance' model for the shaft's process planning is established, as shown in Equation (12). The model assumes that the manufacturing resources of the enterprise will not change in the near future.

$$\begin{aligned} d(m_i, m_j) = & W_F \Delta(F_{m_i}, F_{m_j}) + W_S \Delta(S_{m_i}, S_{m_j}) + W_M \Delta(M_{m_i}, M_{m_j}) \\ & + W_{CT} \Delta(CT_{m_i}, CT_{m_j}) \\ & + (W_C + W_{CP}) \cdot \max(\Delta(C_{m_i}, C_{m_j}), \Delta(CP_{m_i}, CP_{m_j})) \end{aligned} \quad (12)$$

where  $W_F$ ,  $W_S$ ,  $W_M$ ,  $W_{CT}$ ,  $W_C$ , and  $W_{CP}$  represent the weights of the machining feature unit, machining stages, machining machines, machining tools, machining fixtures, and clamping positions, respectively.  $(F_{m_i}, F_{m_j})$ ,  $(S_{m_i}, S_{m_j})$ ,  $(M_{m_i}, M_{m_j})$ ,  $(CT_{m_i}, CT_{m_j})$ ,  $(C_{m_i}, C_{m_j})$ , and  $(CP_{m_i}, CP_{m_j})$  are the machining feature unit, machining stages, machining machines, machining tools, machining fixtures, and clamping positions of machining units  $m_i$  and  $m_j$ .  $\Delta(x, y)$  is the discriminant function between machining units  $m_i$  and  $m_j$ , as shown in Equation (13).

$$\Delta(x, y) = \begin{cases} 0, & x = y \\ 1, & x \neq y \end{cases} \quad (13)$$

#### 3.2. Constraint Conditions and Processing Methods

For the sorting of machining processes, there are generally four basic principles: (a) process the datum surface first; (b) perform roughing before finishing; (c) machine the surface before machining the hole; and (d) machine the primary surface before the secondary surface.

According to the four basic constraints of part processes in the improved ant colony algorithm (IACA), the initial distribution of pheromones for each machining unit is first improved. Based on different initial pheromone distribution weights  $\delta$ , the initial pheromone values for corresponding machining stages are calculated using Equation (5). Secondly, a constraint table  $R_{m_x}$  is added to each machining unit. Assuming that the machining

unit  $m_i$  must be machined before machining unit  $m_j$ , the existence of  $m_j$  is included in the constraint table  $R_{m_i}$ . All operations performed before  $m_j$  are stored in  $R_{m_j}$ . When selecting  $m_j$ , it is necessary to first check whether the constraint table  $R_{m_j}$  is an empty set. If it is an empty set, there is a certain probability of selecting  $m_j$ .

### 3.3. The Specific Steps

Applying IACA to the process planning of large container ship propeller shafts, the algorithm flow is shown in Figure 1, with specific steps as follows:

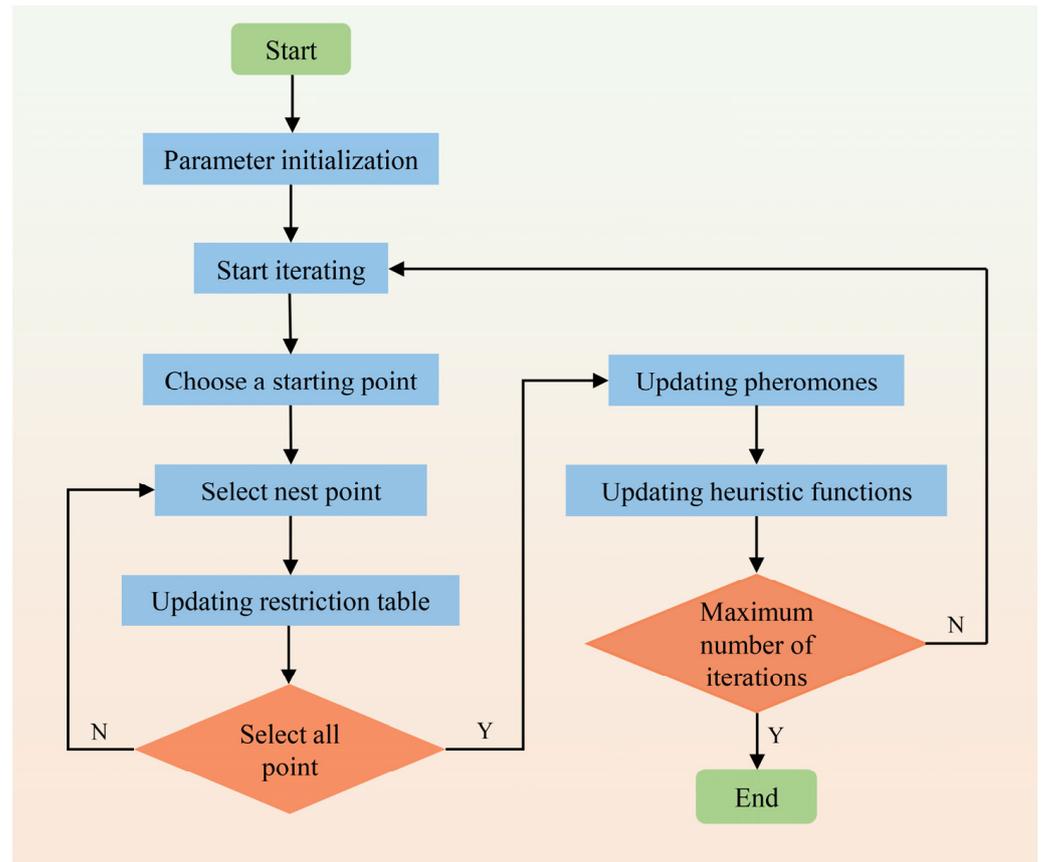


Figure 1. Flowchart of IACA process planning.

Step 1: Initialize parameters. Set parameters such as the number of ants  $m$ , the total pheromone amount  $Q$ , the heuristic factor  $\alpha$ , the and expectation heuristic factor  $\beta$ .

Step 2: Calculate initial pheromones. Calculate the initial pheromones  $\tau_{ij}(0)$  using Equation (5).

Step 3: Select the process. Place ants at points where the constraint table  $R_{m_x}$  is an empty set as starting points. Add this point to the ant's path and remove this point from the constraint tables  $R_{m_x}$  of all processes. Then, calculate the probability of selecting the next process based on whether the constraint table  $R_{m_x}$  is empty and Equation (1).

Step 4: Update pheromones. Perform both local and global pheromone updates and apply the 'reward-penalty system'.

Step 5: Update the expectation heuristic function. Update the expectation heuristic function based on the number of iterations.

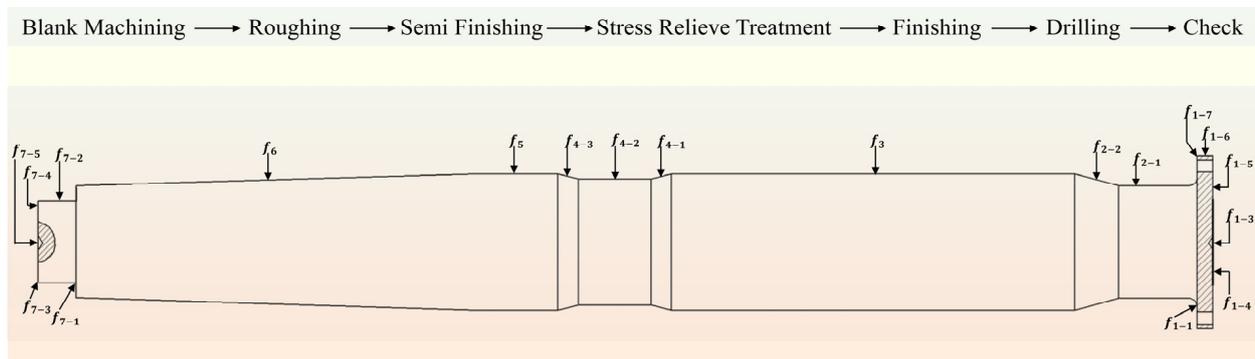
Step 6: Complete the search. Check if the termination conditions are met and calculate the length of the process planning path using Equation (12).

### 4. Model Verification and Analysis

#### 4.1. Application of the Process Planning Model

##### (1) Division of Machining Units for Large Container Ship Propeller Shaft

The machining steps for the large container ship propeller shaft are roughly as follows: blank machining, roughing, semi-finishing, stress relief treatment, finishing, drilling, and checking. To improve the model's practicality and minimize machining errors, we drew a scaled model of the actual-sized shaft, retaining all machining steps. By subdividing the shaft, the machining details of each step could be clearly depicted, as illustrated in Figure 2. The shaft segments mainly included right flange  $f_1$ , transition shaft end  $f_2$ , front common shaft end  $f_3$ , composite shaft end  $f_4$ , rear common shaft end  $f_5$ , tail vertebral body  $f_6$ , and threaded shaft end  $f_7$ . A detailed analysis of the machining steps for each shaft segment is shown in Table 2.



**Figure 2.** The machining process of large container ship propeller shafts and a schematic of the shaft end.

**Table 2.** Detailed machining attributes of each shaft end.

Shaft End Characteristics	Machining Feature Unit	Operation
$f_1$	$f_{1-1}$	Roughing, semi-finishing
	$f_{1-2}$	Boring, reaming
	$f_{1-3}$	Drilling
	$f_{1-4}$	Roughing, semi-finishing
	$f_{1-5}$	Roughing, semi-finishing
	$f_{1-6}$	Roughing, semi-finishing, finishing
	$f_{1-7}$	Roughing, semi-finishing, finishing
$f_2$	$f_{2-1}$	Roughing, semi-finishing, finishing
	$f_{2-2}$	Roughing, semi-finishing, finishing
$f_3$	$f_3$	Roughing, semi-finishing, finishing
$f_4$	$f_{4-1}$	Roughing, semi-finishing, finishing
	$f_{4-2}$	Roughing, semi-finishing, finishing
	$f_{4-3}$	Roughing, semi-finishing, finishing
$f_5$	$f_5$	Roughing, semi-finishing, finishing
$f_6$	$f_6$	Roughing, semi-finishing, finishing
$f_7$	$f_{7-1}$	Roughing
	$f_{7-2}$	Roughing
	$f_{7-3}$	Roughing
	$f_{7-4}$	Roughing, semi-finishing
	$f_{7-5}$	Drilling

##### (2) Initial Information Pheromone Distribution

While adhering to the four basic principles of process planning, it is also important to consider the ‘concentration’ of each machining stage. For example, in Appendix A, if the initial machining selection has the roughing unit  $m_8$  as the positioning reference for  $f_{1-5}$ , at

this point, both the roughing unit  $m_{10}$  and the semi-finishing unit  $m_{11}$  for  $f_{1-6}$  can be chosen simultaneously. This may lead to a lack of concentration at each machining stage. Therefore, a hierarchical initial information pheromone distribution method needs to be adopted, where the initial pheromones for the rough machining stage should be greater than those for the semi-finishing stage, and the initial pheromones for the semi-finishing stage should be greater than those for the finishing stage. In this study, the initial distribution weights for pheromones satisfied the following:  $1 \geq \delta_{Roughing} > \delta_{Semi\ Finishing} > \delta_{Finishing} > 0$ .

(3) Model Weight Coefficients

In the mathematical model for the process planning of large container ship propeller shafts, weights represent the extent to which actual production conditions in the workshop affect the process. Based on the actual workshop conditions, due to the large size of the propeller shaft, changing the machining equipment will inevitably require the workpiece to be re-clamped, which is very time-consuming. Therefore, the weight of machining machines  $W_M$  is the largest. The weights for machining fixtures  $W_C$  and clamping positions  $W_{CP}$  are the second largest. Different machining stages will use different cutting parameters, and the machining time will also vary. Therefore, the weight of machining stages  $W_S$  ranks third. In production, it is preferable to minimize the movement of the cutting tool along the machining path and to machine from one end to the other as much as possible. Therefore, the weight of the machining feature units  $W_F$  ranks fourth. Tool changes during machining have the shortest time, so the weight of machining tools  $W_{CT}$  is the smallest. Specifically,  $W_M > W_C/W_{CP} > W_S > W_F > W_{CT}$ . In this study, the total weight sum was set to 1, and the actual values were determined based on the production line of the enterprise. For example,  $W_M = 0.3, W_C = W_{CP} = 0.2, W_S = 0.15, W_F = 0.1, W_{CT} = 0.05$ .

(4) Machining Constraint Table

Each machining unit has a constraint table  $R_{m_x}$  that stores the machining units that must be performed before the current machining units. Based on the analysis of the machining process for large container ship propeller shafts, constraint tables were set for each machining unit, as shown in Table 3. Each pair forms a column, where ‘machining units’ represent the machining units for large container ship propeller shafts, totaling 45 steps, and ‘constraint table’ signifies the constraint units of the machining units. Upon the completion of machining a constraint unit from the constraint table, the corresponding machining units will then have a certain probability of being selected.  $m_8$  and  $m_{43}$  are used to machine the positioning reference holes on the left and right ends of the propeller shaft, so  $m_8$  and  $m_{43}$  have no constraints in front of them and can be used as the starting positions for machining.

The constraint table can guide the choices made by ants. First, select the initial machining position in  $m_8$  or  $m_{43}$ . If  $m_8$  is chosen, remove  $m_8$  from the constraint tables of various machining units and bring out the machining units with empty constraint tables, which are  $m_5, m_6, m_9$  and  $m_{10}$ . Then, make a choice based on the calculated probabilities and continue this process until all machining steps are completed. As shown in Figure 3.

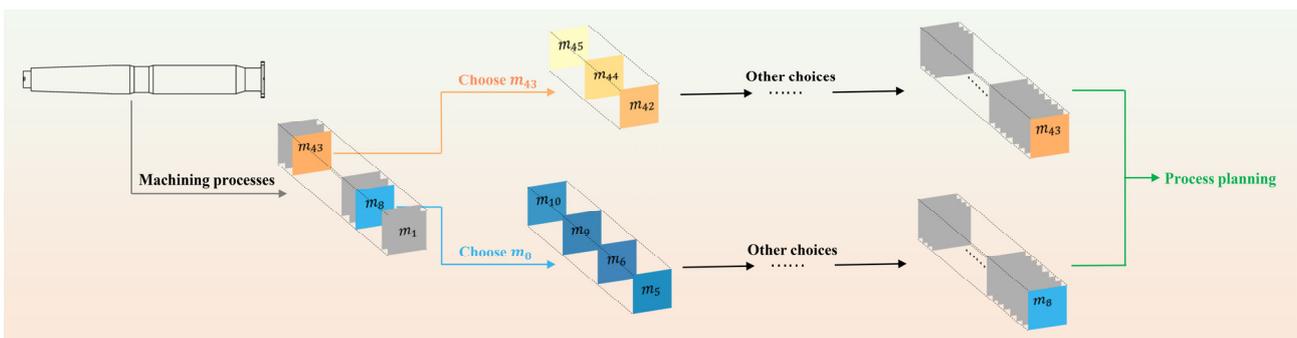


Figure 3. The role of constraint tables.

Table 3. Constraint table for each machining unit.

Machining Units ( $m_x$ )	Constraint Table ( $R_{m_x}$ )	Machining Units ( $m_x$ )	Constraint Table ( $R_{m_x}$ )	Machining Units ( $m_x$ )	Constraint Table ( $R_{m_x}$ )
$m_1$	$m_5, m_8$	$m_{16}$	$m_1$	$m_{31}$	$m_{28}$
$m_2$	$m_1$	$m_{17}$	$m_{16}$	$m_{32}$	$m_{31}$
$m_3$	$m_{39}$	$m_{18}$	$m_{17}$	$m_{33}$	$m_{32}$
$m_4$	$m_3$	$m_{19}$	$m_{16}$	$m_{34}$	$m_{31}$
$m_5$	$m_8$	$m_{20}$	$m_{19}$	$m_{35}$	$m_{34}$
$m_6$	$m_8$	$m_{21}$	$m_{20}$	$m_{36}$	$m_{35}$
$m_7$	$m_6$	$m_{22}$	$m_{19}$	$m_{37}$	$m_{34}$
$m_8$	/	$m_{23}$	$m_{22}$	$m_{38}$	$m_{37}$
$m_9$	$m_8$	$m_{24}$	$m_{23}$	$m_{39}$	$m_{38}$
$m_{10}$	$m_8$	$m_{25}$	$m_{22}$	$m_{40}$	$m_{37}$
$m_{11}$	$m_{10}$	$m_{26}$	$m_{25}$	$m_{41}$	$m_{42}$
$m_{12}$	$m_{11}$	$m_{27}$	$m_{26}$	$m_{42}$	$m_{43}$
$m_{13}$	$m_1, m_{10}$	$m_{28}$	$m_{25}$	$m_{43}$	/
$m_{14}$	$m_{13}$	$m_{29}$	$m_{28}$	$m_{44}$	$m_{43}$
$m_{15}$	$m_{14}$	$m_{30}$	$m_{29}$	$m_{45}$	$m_{43}$

In each pair of columns, the former represents the machining units for large container ship propeller shafts, while the latter represents the constraint units of the current machining units.

#### 4.2. Parameter Optimization and Selection

The ant colony algorithm is primarily influenced by the calculation probability of selecting path points and the method of updating pheromones. From Equations (1) and (3), it can be seen that the ant colony algorithm has three main parameters: the heuristic factor  $\alpha$ , the expectation heuristic factor  $\beta$ , and the volatilization coefficient  $\rho$ . The range of values for each parameter is shown in Table 4.

Table 4. Ranges of parameter values.

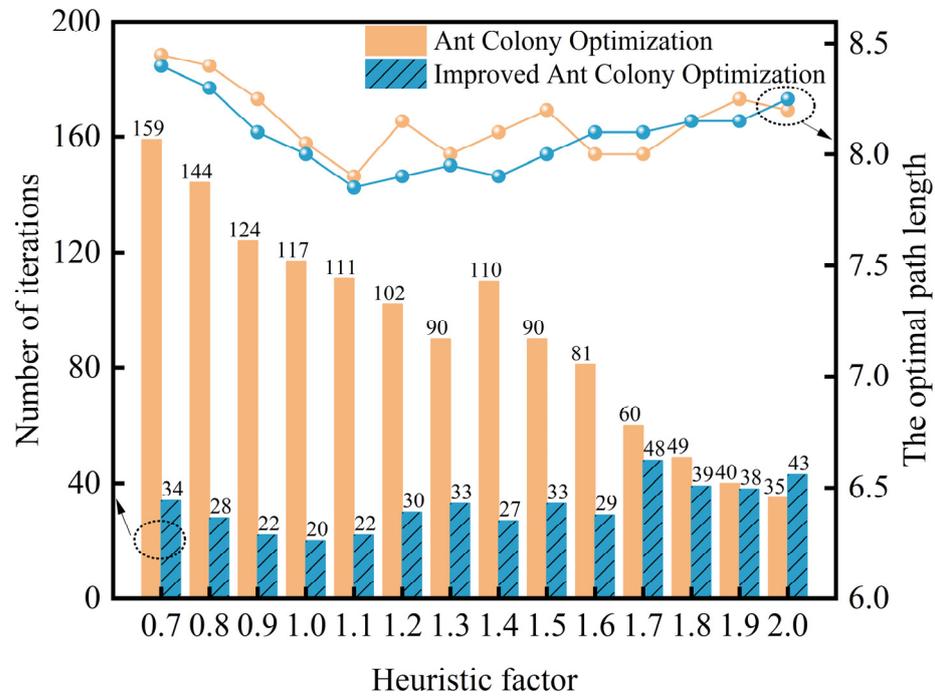
Parameter	Value Ranges
$\alpha$	(0, 2]
$\beta$	[4, 7]
$\rho$	(0, 1)

To date, there is no comprehensive analysis method available to directly determine the optimal parameter combination. Therefore, based on the practical issues of large container ship propeller shaft process planning and the actual production circumstances of the enterprise, we employed a single-factor control variable method. By systematically altering one parameter at a time, we analyzed the individual impacts of each parameter on algorithmic results, thereby determining the weighted optimal parameter combination. The modeling environment was as follows: Windows 11 64-bit; processor: Intel (R) Core (TM) i7-8750H; clock speed: 2.20 GHz; memory: 8 GB; algorithm simulation software: PyCharm 2023.3.

The initial settings of parameters for the model simulation were as follows:  $Ant\_number = 100$ ,  $Itermax = 200$ ,  $Q = 100$ ,  $\beta = 0.6$ ,  $\rho = 0.3$ . The algorithm's convergence iterations and the best path length were tested for  $\alpha$  values ranging from 0.7 to 2.0. The simulation experimental results are shown in Figure 4.

The value of the heuristic factor  $\alpha$  has a significant impact on the ant colony algorithm. A lower  $\alpha$  weakens the effect of pheromones, promoting exploration of new paths but delaying convergence. Conversely, a higher  $\alpha$  strengthens path dependence, reduces exploration of new routes, and facilitates local optima. From Figure 4, it can be observed that when the information pheromone factor  $\alpha$  is small, both algorithms have poor search capabilities and are prone to getting stuck in local optima. As  $\alpha$  gradually increases, the search capabilities of both algorithms improve. Moreover, the improved ant colony

algorithm not only maintains high search capabilities but also achieves faster convergence compared to the traditional ant colony algorithm. When  $\alpha$  is in the range of  $[1.0, 1.2]$ , both algorithms exhibit fast convergence and good search capabilities. Specifically, when  $\alpha = 1.1$ , the search capabilities and convergence speed are excellent, with the improved ant colony algorithm converging in 22 iterations and achieving a global optimal path length of 7.85.



**Figure 4.** Relationship between the heuristic factor, the number of iterations, and the optimal path length.

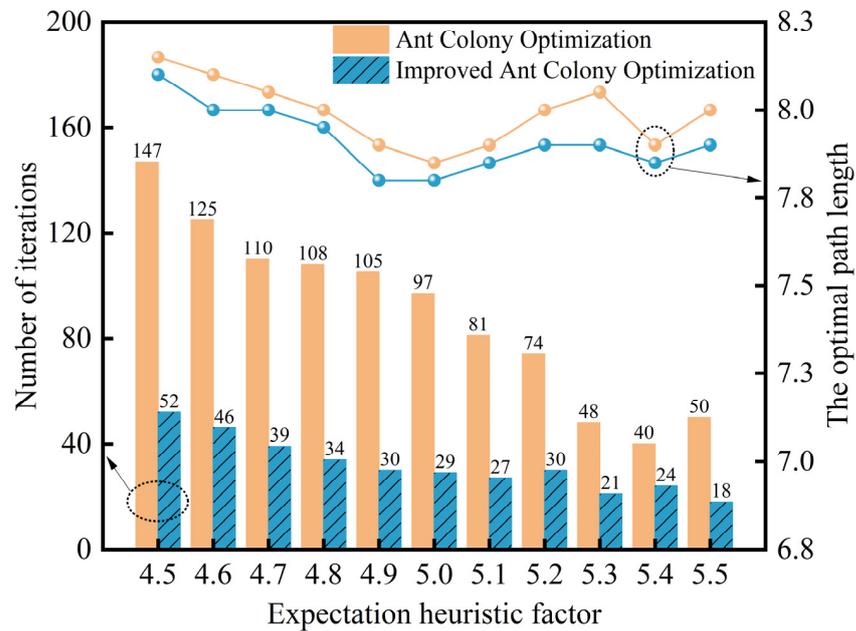
In the same environment, the parameters were set as follows:  $Ant\_number = 100$ ,  $Itermax = 200$ ,  $Q = 100$ ,  $\alpha = 1.1$ ,  $\rho = 0.3$ . The algorithm was tested for different values of the expectation heuristic factor  $\beta$  ranging from 4.5 to 5.5, and the results are shown in Figure 5.

A smaller value of the expectation heuristic factor  $\beta$  can lead to an unclear direction in the ant colony algorithm, prolonging the search time and resulting in slow convergence. On the other hand, a larger  $\beta$  may lead to premature convergence to local optimal solutions. From Figure 5, it can be observed that when  $\beta$  is small, both algorithms have an unclear search direction and slow convergence. As  $\beta$  gradually increases, both algorithms converge faster but may produce suboptimal search results, possibly getting stuck in local optima. When  $\beta$  is in the range of  $[4.9, 5.1]$ , both algorithms exhibit strong guidance during the search and achieve faster convergence. Specifically, when  $\beta = 5.0$ , the improved ant colony algorithm converges in 29 iterations and achieves a global optimal path length of 7.8.

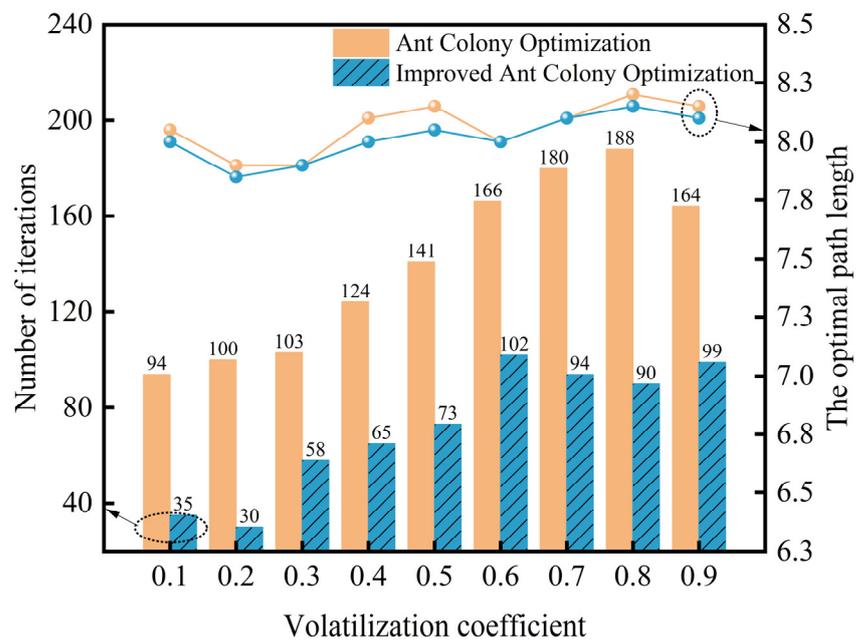
In the same experimental setup with  $\alpha = 1.1$  and  $\beta = 5$ , the algorithm's performance was tested for different values of the volatilization coefficient  $\rho$  ranging from 0.1 to 0.9, and the results are shown in Figure 6.

A smaller value of the volatilization coefficient  $\rho$  leads to slow pheromone evaporation, increasing the risk of getting stuck in local optimal solutions. Conversely, a larger  $\rho$  causes rapid pheromone evaporation, affecting the utilization of previous search experiences and slowing down algorithm convergence. From Figure 6, it can be observed that when  $\rho$  is too small, both algorithms produce suboptimal search results, indicating that they are trapped in local optima. As  $\rho$  gradually increases, the search capabilities of both algorithms deteriorate, and the convergence speed decreases. When  $\rho$  is in the range of  $[0.2, 0.3]$ , both algorithms exhibit good search capabilities and convergence speed. Specifically, when

$\rho = 0.2$ , the improved ant colony algorithm converges in 30 iterations and achieves a global optimal path length of 7.8.



**Figure 5.** Relationship between the expectation heuristic factor, the number of iterations, and the optimal path length.



**Figure 6.** Relationship between the volatilization coefficient, the number of iterations, and the optimal path length.

### 4.3. Comparison with Other Algorithms

When  $\alpha \in [1.0, 1.2]$ ,  $\beta \in [5.0, 5.2]$ , and  $\rho \in [0.2, 0.3]$ , the performance of the ant colony algorithm is superior. To validate the effectiveness and superiority of this algorithm, model simulation experiments were conducted with the following improved ant colony algorithm parameters:  $Ant\_number = 100$ ,  $Itermax = 200$ ,  $Q = 100$ ,  $\alpha = 1.1$ ,  $\beta = 5.0$ ,  $\rho = 0.2$ . To eliminate randomness, ACA, IACA, genetic algorithm (GA), and particle swarm optimization (PSO) were each tested ten times. The average optimal path value,

average number of iterations, and average runtime for each algorithm over the ten tests were calculated and are recorded in Table 5. It can be observed that, compared to GA, ACA and IACA exhibit significant advantages in convergence speed and final convergence results, indicating that ACA and IACA are relatively simpler and more controllable in adjusting algorithm parameters. Compared to PSO, ACA and IACA demonstrate better iteration stability, suggesting that ACA and IACA are less prone to being trapped in local optima and thus escaping the iteration process. In comparison to ACA, IACA achieves further improvements in convergence speed and iteration stability while also avoiding the contradiction between convergence speed and susceptibility to local optima. Additionally, to compare the initial search capabilities and convergence speeds of ACA and IACA, we further analyzed the optimal iteration results of the two algorithms, as shown in Figure 7.

Table 5. Comparison of process planning.

Algorithm	Optimal Path Length	Optimal Path Mean Length	Average Number of Iterations	Average Run Time
ACA	7.8	8.06	136	108 s
IACA	7.8	7.8	29	23 s
GA	8.2	8.4	153	125 s
PSO	9.0	9.6	74	58 s

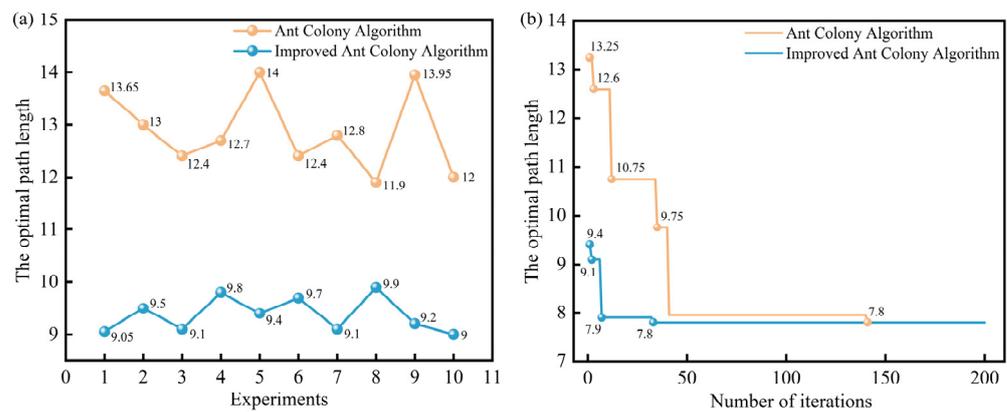


Figure 7. (a) Optimal path values for the first iteration; (b) optimal path convergence curve.

From Figure 7 and Table 5, it can be observed that in the ten tests conducted, both ACA and IACA achieve an optimal path value of 7.8. However, the latter exhibits stronger stability in terms of the average optimal path value. The initial iteration’s optimal path value for ACA differs from the average optimal path value by approximately 48% to 74%. Moreover, the average iteration counts and average running times for the ACA are relatively high, indicating that the initial search of the ACA is highly blind and lacks strong guidance for subsequent iterations, resulting in slow convergence. In contrast, the initial iteration’s optimal path value for the IACA is significantly lower than that of ACA, differing from the average optimal path value by approximately 15% to 27%. Additionally, the average iteration count decreases by about 79%, and the average running time decreases by approximately 83%. This suggests that the improved algorithm avoids the blind search in the initial phase, leading to improved convergence speed.

The optimized process routes of the two algorithms were compared with the existing enterprise process route, as shown in Table 6. It can be concluded that, in the current machining process route for large container ship propeller shafts, tool changes occur 11 times, workpiece clamping is changed five times, and the machine tool is changed once. The optimized machining process route reduces tool changes to 8 times, maintains workpiece clamping changes at five times, and machine tool changes at 1 time. Moreover, it reduces two workpiece rotations and includes two additional drilling operations in the optimized route. However, during detailed machining, the optimization result of ACA

involves tool movement from both ends towards the center, while IACA adopts a tool path that starts from one end to the other and then reverses at the end of machining. This reduces unnecessary tool movements during machining, improves the continuity of the machining path, and makes the entire machining process more compact and efficient.

**Table 6.** Large container ship propeller shaft process planning.

Large Container Ship Propeller Shaft Process Planning	
Existing	Roughing $f_{7-4}$ , Drilling $f_{7-5}$ , Roughing $f_{7-3}$ , Roughing $f_{7-2}$ , Roughing $f_{1-5}$ , Roughing $f_{1-4}$ , Drilling $f_{1-3}$ , Roughing $f_{1-6}$ , Roughing $f_{1-1}$ , Roughing $f_{1-7}$ , Roughing $f_{2-1}$ , Roughing $f_{2-2}$ , Roughing $f_3$ , Roughing $f_{4-1}$ , Roughing $f_{4-2}$ , Roughing $f_{4-3}$ , Roughing $f_5$ , Roughing $f_6$ , Roughing $f_{7-1}$ , Semi-finishing $f_{1-5}$ , Semi-finishing $f_{1-4}$ , Drilling $f_{1-3}$ , Semi-finishing $f_{1-6}$ , Semi-finishing $f_{1-1}$ , Semi-finishing $f_{1-7}$ , Semi-finishing $f_{2-1}$ , Semi-finishing $f_{2-2}$ , Semi-finishing $f_3$ , Semi-finishing $f_{4-1}$ , Semi-finishing $f_{4-2}$ , Semi-finishing $f_{4-3}$ , Semi-finishing $f_5$ , Semi-finishing $f_6$ , Semi-finishing $f_{7-4}$ , Drilling $f_{7-5}$ , Finishing $f_{1-6}$ , Finishing $f_{1-7}$ , Finishing $f_{2-1}$ , Finishing $f_{2-2}$ , Finishing $f_3$ , Finishing $f_{4-1}$ , Finishing $f_{4-2}$ , Finishing $f_{4-3}$ , Finishing $f_5$ , Finishing $f_6$ , Boring $f_{1-2}$ , Reaming $f_{1-2}$
ACA	Roughing $f_{7-4}$ , Semi-finishing $f_{7-4}$ , Drilling $f_{7-5}$ , Roughing $f_{7-3}$ , Roughing $f_{7-2}$ , Roughing $f_{1-5}$ -Roughing $f_{1-4}$ , Semi-finishing $f_{1-4}$ , Semi-finishing $f_{1-5}$ , Drilling $f_{1-3}$ , Roughing $f_{1-6}$ , Roughing $f_{1-1}$ , Roughing $f_{2-1}$ , Roughing $f_{1-7}$ , Roughing $f_{2-2}$ -Roughing $f_3$ -Roughing $f_{4-1}$ -Roughing $f_{4-2}$ -Roughing $f_{4-3}$ , Roughing $f_5$ , Roughing $f_6$ , Roughing $f_{7-1}$ , Semi-finishing $f_{1-1}$ , Semi-finishing $f_{1-7}$ , Semi-finishing $f_3$ , Semi-finishing $f_{4-2}$ , Semi-finishing $f_{2-2}$ , Semi-finishing $f_{2-1}$ , Semi-finishing $f_{1-6}$ , Semi-finishing $f_5$ , Semi-finishing $f_{4-1}$ , Semi-finishing $f_{4-3}$ , Semi-finishing $f_6$ , Finishing $f_6$ , Finishing $f_{2-1}$ , Finishing $f_3$ , Finishing $f_{2-2}$ , Finishing $f_5$ , Finishing $f_{1-6}$ , Finishing $f_{1-7}$ , Finishing $f_{4-2}$ , Finishing $f_{4-3}$ , Finishing $f_{4-1}$ , Boring $f_{1-2}$ , Reaming $f_{1-2}$
IACA	Roughing $f_{7-4}$ , Semi-finishing $f_{7-4}$ , Drilling $f_{7-5}$ , Roughing $f_{7-3}$ , Roughing $f_{7-2}$ , Roughing $f_{1-5}$ -Roughing $f_{1-4}$ , Semi-finishing $f_{1-4}$ , Semi-finishing $f_{1-5}$ , Drilling $f_{1-3}$ , Roughing $f_{1-6}$ , Roughing $f_{1-1}$ , Roughing $f_{1-7}$ , Roughing $f_{2-1}$ , Roughing $f_{2-2}$ -Roughing $f_3$ -Roughing $f_{4-1}$ -Roughing $f_{4-2}$ -Roughing $f_{4-3}$ , Roughing $f_5$ , Roughing $f_6$ , Roughing $f_{7-1}$ , Semi-finishing $f_6$ , Semi-finishing $f_5$ , Semi-finishing $f_{4-3}$ , Semi-finishing $f_{4-2}$ , Semi-finishing $f_{4-1}$ , Semi-finishing $f_3$ , Semi-finishing $f_{2-2}$ , Semi-finishing $f_{2-1}$ , Semi-finishing $f_{1-7}$ , Semi-finishing $f_{1-1}$ , Semi-finishing $f_{1-6}$ , Finishing $f_{1-6}$ , Finishing $f_{1-7}$ , Finishing $f_{2-1}$ , Finishing $f_{2-2}$ , Finishing $f_3$ , Finishing $f_{4-1}$ , Finishing $f_{4-2}$ , Finishing $f_{4-3}$ , Finishing $f_5$ , Finishing $f_6$ , Boring $f_{1-2}$ , Reaming $f_{1-2}$

### 5. Benefit Analysis of Process Planning

#### 5.1. Working Time Quota Analysis

The working time quota, also known as the time quota, refers to the time required to complete the processing of parts according to the product processing process in a certain technical state and production organization mode. After actual investigation, the work time quota of this enterprise is generally composed of three parts: basic processing time, auxiliary processing time, and worker rest time. The basic processing time is the time required for dimension processing of parts on the machine tool, mainly including turning time and drilling, boring time, etc. The auxiliary processing time is the time spent on various auxiliary operations in the processing of parts, mainly including part measurement time, part turning and loading and unloading time, fixture and tool replacement time, the time for adjusting and cleaning the machine tool, etc.

In actual production, compared to the existing processing technology route of the enterprise, when using the optimized processing technology route by IACA for production, the number of parts turning and drilling is reduced, and the time for workers to move back and forth at both ends of the axis is reduced when transitioning between rough machining and semi-precision machining and between semi-precision machining and precision machining. Through the recording of actual production times, the fundamental machining times during production using optimized machining process routes are presented in Table 7. As shown in Table 7, the machining times for each machining stages of various segments of the large container ship propeller shaft were statistically analyzed, and the machining times

for each machining stage of the shaft were summed to obtain the fundamental machining times for each machining stage of the shaft. Concurrently, practical machining tests were conducted for all three machining schemes, and the machining times for each scheme were recorded, as shown in Table 8.

**Table 7.** The basic processing time.

Shaft End Characteristics		Machining Stages			
		Roughing (min)	Semi-Finishing (min)	Finishing (min)	Boring and Reaming (min)
$f_1$	$f_{1-3}$	17.7	/	/	
	$f_{1-1}f_{1-6}$	120	150	/	
	$f_{1-4}$	76	95	/	/
	$f_{1-5}$	30	37.5	75	
	$f_{1-2}$		/	/	900
	$f_2$	81	101.3	202.5	
	$f_3$	162.5	203.1	406.5	
	$f_4$	595	743.8	1487.5	
	$f_5$	230	287.5	575	
	$f_6$	173	216.3	432.5	/
$f_7$	$f_{7-5}$	17.7	/	/	
	$f_{7-4}$	31.5	39.4	/	
	$f_{7-1}f_{7-2}f_{7-3}$	33.5	/	/	
Basic Processing Time		1567.9	1873.9	3178.8	900

**Table 8.** Comparison of working time quotas for the three options.

	Basic Processing Time (min)	Auxiliary Processing Time (min)
Existing	7580.0	1148.4
ACA	7520.6	752.1
IACA	7520.6	712.1

From Tables 7 and 8, it can be concluded that in the basic processing time, both the traditional ant colony algorithm and the improved ant colony algorithm optimized processing routes reduce the number of drilling operations for  $f_1$  and  $f_7$  by two times, resulting in a current drilling time of 35.4 min, saving approximately 59.4 min compared to the existing enterprise process route. In the auxiliary processing time, the optimized route reduces part turning by two times, saving approximately 300 min, and also reduces tool changes by three times, saving about 96.3 min. The process route optimized by the traditional ant colony algorithm saves about 455.7 min compared to the existing enterprise process route. IACA further optimizes the tool path, effectively reducing tool movement time, saving approximately 40 min again. Therefore, IACA saves about 415.7 min compared to the existing enterprise process route.

5.2. Process Cost Analysis

Process cost generally refers to the manufacturing cost closely associated with the production stage and the process. It consists of variable costs and fixed costs. Variable costs mainly include labor wages, tool consumption expenses, etc., while fixed costs mainly encompass equipment depreciation, maintenance fees, etc. When choosing among several technically equivalent process options, it is common to compare the process costs of each option, with the one with the lowest process cost considered the optimal solution.

The processing equipment used for large container ship propeller shafts are general-purpose machine tools. Additionally, these propeller shafts are produced as single pieces, so there is no need for specialized fixtures. Therefore, four cost factors, namely, machine

operator wages, depreciation of general-purpose machine tools, tool maintenance costs, and depreciation of general-purpose fixtures, were considered.

Machine operator wages:

$$C_M = (T_d Z_h / 60) \times (1 + \alpha) \tag{14}$$

where  $T_d$  represents the total time for a single unit,  $Z_h$  is the unit time wage for machine tool workers, and  $\alpha$  is the additional wage rate for the compensation reward mechanism.

General machine tool depreciation cost:

$$C_{M_d} = P_M R_M T_d / 6000 F \omega_M \tag{15}$$

where  $P_M$  is the machine tool price,  $R_M$  is the machine tool depreciation rate,  $F$  is the machine tool working time, and  $\omega_M$  is the machine tool load factor.

Tool maintenance cost:

$$C_{CT} = n P_o T_j / T_e (n + 1) \tag{16}$$

where  $n$  is the number of times a tool can be sharpened,  $P_o$  is the cost of sharpening the tool once,  $T_j$  is the basic machining time, and  $T_e$  is the durability of the tool.

General fixture depreciation cost:

$$C_{C_d} = P_C R_C T_d / 6000 F \omega_C \tag{17}$$

where  $P_C$  is the fixture price,  $R_C$  is the fixture depreciation rate,  $\omega_C$  is the utilization rate of the fixture in production, and  $F$  is the fixture working time similar to the machine tool. As shown in Table 9.

**Table 9.** Value of each process cost item.

Notation	Value	Unit	Notation	Value	Unit
$Z_h$	80	CNY/h	$n$	5	
$\alpha$	10	%	$P_o$	3	CNY/times
$P_M$	2,600,000	CNY	$T_e$	100	min
$R_M$	10	%	$P_C$	40,000	CNY
$F$	1050	min	$R_C$	33	%
$\omega_M$	80	%	$\omega_C$	100	%

Based on the calculations from the above formulas, the economic comparison of the existing enterprise process route and the two optimized algorithms is presented in Table 10. It can be observed that all the process costs for the optimized process routes are lower than the current enterprise solution. Specifically, when comparing ACA and IACA in terms of machine tool worker wages, the former decreases by CNY 668.4, while the latter decreases by CNY 727.1. The depreciation cost of general machine tools decreases by CNY 23.5 for the former and CNY 25.6 for the latter. Both algorithms result in a reduction of CNY 1.5 in tool maintenance costs. As for the depreciation cost of general fixtures, the former decreases by CNY 1 and the latter by CNY 1.1. In total, the process cost decreases by CNY 694.4 for the former and CNY 755.3 for the latter. The improved ant colony algorithm reduces the cost by an additional CNY 60.9 compared to the traditional ant colony algorithm.

**Table 10.** Comparison of the economics of the three options.

	Machine Operator Wages (CNY)	General Machine Tool Depreciation Cost (CNY)	Tool Maintenance Cost (CNY)	General Fixture Depreciation Cost (CNY)
Existing	12,801.7	450.3	189.5	18.3
ACA	12,133.3	426.8	188.0	17.3
IACA	12,074.6	424.7	188.0	17.2

## 6. Conclusions

In the face of a market characterized by complex multi-processes and extensive customization of products, it is crucial to respond quickly and address the challenges of process planning for intricate components or products. This study, after analyzing the ACA's application in solving process planning problems, focused on addressing issues such as the algorithm's blind initial search, contradictory convergence speed, and susceptibility to local optimal solutions. Improvements were made to the initial pheromone distribution, heuristic function update, and pheromone update methods in response to the challenges posed by the traditional ant colony algorithm. The result was IACA, which was then applied to the process planning of large container ship propeller shafts. The main conclusions are as follows:

1. Improved initial distribution of information pheromones: Based on the fundamental principles of part process planning, a hierarchical distribution of pheromones is proposed, reducing the initial search error by approximately 40%. This eliminates the initial search's randomness and provides better guidance for subsequent iterations.

2. Analysis of the role of heuristic functions in the ant colony algorithm: A mathematical model for calculating the heuristic function is introduced by comparing the similarities and differences between machining units as a measure of distance. The introduction of a decay factor  $\zeta$  reduces the influence of the heuristic function on path selection in the later stages, leading to a roughly 79% improvement in convergence speed.

3. Improved update mechanism for information pheromones: A reward–penalty system is introduced, where ants that follow the optimal path are 'rewarded', while those following the worst path are 'punished'. This accelerates the algorithm's convergence speed while keeping the pheromone levels within specified bounds, preventing the algorithm from getting stuck in local optima and experiencing premature stagnation.

4. Integration of process constraints: Real-world production constraints are considered, and constraint tables are introduced for each machining unit based on the four fundamental principles of part process planning and the actual production capabilities of the enterprise. This allows the algorithm to perform process route iteration optimization in accordance with the basic machining principles.

5. Benefit analysis: In practical production, the single-piece processing time for large container ship propeller shafts is reduced by approximately 415.7 min, and the process cost is reduced by approximately CNY 755.3. This demonstrates the feasibility and effectiveness of applying the improved ant colony algorithm to the optimization of ship propulsion shaft machining processes.

**Author Contributions:** Methodology, G.D. and N.M.; software, G.D.; writing—original draft preparation, G.D.; writing—review and editing, N.M.; validation, G.D., H.M. and Y.B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Natural Science Foundation of Shandong Province (ZR2023ME146) and the Postgraduate Education Joint Cultivation Base Construction Projects, Ocean University of China (HDYJ23005).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** Author Hongkui Ma was employed by the company Qingdao Hiron Commercial Cold Chain Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Appendix A

Details of each machining unit of a propeller shaft for a large container ship.

Machining Unit	Machining Feature Unit	Machining Stages	Machining Tools	Machining Fixtures	Clamping Positions	Machining Machines	
$m_1$	$f_{1-1}$	Roughing	Cutting tool 4225	Four-jaw chuck, lathe	$f_{1-3}f_{7-2}$	Turning machines	
$m_2$	$f_{1-1}$	Semi-finishing	Cutting tool 4225		$f_{1-3}f_{7-2}$		
$m_3$	$f_{1-2}$	Boring	Boring tool	Self-made special fixture	$f_3f_5$	Boring machines	
$m_4$	$f_{1-2}$	Reaming	Reaming tool		$f_3f_5$		
$m_5$	$f_{1-3}$	Drilling	Drilling tool	Four-jaw chuck, center rest	$f_3f_{7-2}$		
$m_6$	$f_{1-4}$	Roughing	Cutting tool 4225		$f_3f_{7-2}$		
$m_7$	$f_{1-4}$	Semi-finishing	Cutting tool 4225		$f_3f_{7-2}$		
$m_8$	$f_{1-5}$	Roughing	Cutting tool 4225		$f_3f_{7-2}$		
$m_9$	$f_{1-5}$	Semi-finishing	Cutting tool 4225		$f_3f_{7-2}$		
$m_{10}$	$f_{1-6}$	Roughing	Cutting tool 4225		$f_{1-3}f_{7-2}$		
$m_{11}$	$f_{1-6}$	Semi-finishing	Cutting tool 4225		$f_{1-3}f_{7-2}$		
$m_{12}$	$f_{1-6}$	Finishing	Cutting tool 4025		$f_{1-3}f_{7-2}$		
$m_{13}$	$f_{1-7}$	Roughing	Cutting tool 4225		$f_{1-3}f_{7-2}$		
$m_{14}$	$f_{1-7}$	Semi-finishing	Cutting tool 4225		$f_{1-3}f_{7-2}$		
$m_{15}$	$f_{1-7}$	Finishing	Cutting tool 4025		$f_{1-3}f_{7-2}$		
$m_{16}$	$f_{2-1}$	Roughing	Cutting tool 4225	$f_{1-3}f_{7-2}$			
$m_{17}$	$f_{2-1}$	Semi-finishing	Cutting tool 4225	$f_{1-3}f_{7-2}$			
$m_{18}$	$f_{2-1}$	Finishing	Cutting tool 4025	$f_{1-3}f_{7-2}$			
$m_{19}$	$f_{2-2}$	Roughing	Cutting tool 4225	$f_{1-3}f_{7-2}$			
$m_{20}$	$f_{2-2}$	Semi-finishing	Cutting tool 4225	$f_{1-3}f_{7-2}$			
$m_{21}$	$f_{2-2}$	Finishing	Cutting tool 4025	$f_{1-3}f_{7-2}$			
$m_{22}$	$f_3$	Roughing	Cutting tool 4225	$f_{1-3}f_{7-2}$			
$m_{23}$	$f_3$	Semi-finishing	Cutting tool 4225	$f_{1-3}f_{7-2}$		Turning machines	
$m_{24}$	$f_3$	Finishing	Cutting tool 4025	$f_{1-3}f_{7-2}$			
$m_{25}$	$f_{4-1}$	Roughing	Cutting tool 4225	Four-jaw chuck, lathe		$f_{1-3}f_{7-2}$	
$m_{26}$	$f_{4-1}$	Semi-finishing	Cutting tool 4225			$f_{1-3}f_{7-2}$	
$m_{27}$	$f_{4-1}$	Finishing	Cutting tool 4025		$f_{1-3}f_{7-2}$		
$m_{28}$	$f_{4-2}$	Roughing	Cutting tool 4225		$f_{1-3}f_{7-2}$		
$m_{29}$	$f_{4-2}$	Semi-finishing	Cutting tool 4225		$f_{1-3}f_{7-2}$		
$m_{30}$	$f_{4-2}$	Finishing	Cutting tool 4025		$f_{1-3}f_{7-2}$		
$m_{31}$	$f_{4-3}$	Roughing	Cutting tool 4225		$f_{1-3}f_{7-2}$		
$m_{32}$	$f_{4-3}$	Semi-finishing	Cutting tool 4225		$f_{1-3}f_{7-2}$		
$m_{33}$	$f_{4-3}$	Finishing	Cutting tool 4025		$f_{1-3}f_{7-2}$		
$m_{34}$	$f_5$	Roughing	Cutting tool 4225		$f_{1-3}f_{7-2}$		
$m_{35}$	$f_5$	Semi-finishing	Cutting tool 4225		$f_{1-3}f_{7-2}$		
$m_{36}$	$f_5$	Finishing	Cutting tool 4025	$f_{1-3}f_{7-2}$			
$m_{37}$	$f_6$	Roughing	Cutting tool 4225	$f_{1-3}f_{7-2}$			
$m_{38}$	$f_6$	Semi-finishing	Cutting tool 4225	$f_{1-3}f_{7-2}$			
$m_{39}$	$f_6$	Finishing	Cutting tool 4025	$f_{1-3}f_{7-2}$			
$m_{40}$	$f_{7-1}$	Roughing	Cutting tool 4225	$f_{1-3}f_{7-2}$			
$m_{41}$	$f_{7-2}$	Roughing	Cutting tool 4225	$f_{1-6}f_5$			

Machining Unit	Machining Feature Unit	Machining Stages	Machining Tools	Machining Fixtures	Clamping Positions	Machining Machines
$m_{42}$	$f_{7-3}$	Roughing	Cutting tool 4225	Four-jaw chuck, center rest	$f_{1-6f5}$	Turning machines
$m_{43}$	$f_{7-4}$	Roughing	Cutting tool 4225		$f_{1-6f5}$	
$m_{44}$	$f_{7-4}$	Semi-finishing	Cutting tool 4025		$f_{1-6f5}$	
$m_{45}$	$f_{7-5}$	Drilling	Drilling tool		$f_{1-6f5}$	

## References

- Zhang, C.; Chen, Y.; Chen, H.; Chong, D. Industry 4.0 and its Implementation: A Review. *Inf. Syst. Front.* **2021**. [CrossRef]
- Xu, X.; Lu, Y.; Vogel-Heuser, B.; Wang, L. Industry 4.0 and Industry 5.0—Inception, conception and perception. *J. Manuf. Syst.* **2021**, *61*, 530–535. [CrossRef]
- Zhou, J.; Camba, J.D. Computer-aided process planning in immersive environments: A critical review. *Comput. Ind.* **2021**, *133*, 103547. [CrossRef]
- Arinez, J.F.; Chang, Q.; Gao, R.X.; Xu, C.; Zhang, J. Artificial Intelligence in Advanced Manufacturing: Current Status and Future Outlook. *J. Manuf. Sci. Eng.* **2020**, *142*, 110804. [CrossRef]
- Trstenjak, M.; Opetuk, T.; Cajner, H.; Tosanovic, N. Process Planning in Industry 4.0—Current State, Potential and Management of Transformation. *Sustainability* **2020**, *12*, 5878. [CrossRef]
- Zhang, Q.; Xiong, S. Routing optimization of emergency grain distribution vehicles using the immune ant colony optimization algorithm. *Appl. Soft Comput.* **2018**, *71*, 917–925. [CrossRef]
- Perez-Carabaza, S.; Besada-Portas, E.; Lopez-Orozco, J.A.; De La Cruz, J.M. Ant colony optimization for multi-UAV minimum time search in uncertain domains. *Appl. Soft Comput.* **2018**, *62*, 789–806. [CrossRef]
- Xiang, X.; Qiu, J.; Xiao, J.; Zhang, X. Demand coverage diversity based ant colony optimization for dynamic vehicle routing problems. *Eng. Appl. Artif. Intell.* **2020**, *91*, 103582. [CrossRef]
- Deng, T.; Li, Y.; Liu, X. An inexact subgraph matching algorithm for subpart retrieval in NC process reuse. *J. Manuf. Syst.* **2023**, *67*, 410–423. [CrossRef]
- Huang, B.; He, K.; Huang, R.; Zhang, F.; Li, X.; Zhang, S. Efficient NC process scheme generation method based on reusable macro and micro process fusion. *Int. J. Adv. Manuf. Technol.* **2022**, *120*, 2517–2535. [CrossRef]
- Kowalski, M.; Zawadzki, P. Intelligent manufacturing templates for CNC machine programming. *J. Phys. Conf. Ser.* **2022**, *2198*, 012041. [CrossRef]
- Zhang, Y.; Zhang, S.; Huang, R.; Huang, B.; Liang, J.; Zhang, H.; Wang, Z. Combining deep learning with knowledge graph for macro process planning. *Comput. Ind.* **2022**, *140*, 103668. [CrossRef]
- Ma, H.; Zhou, X.; Liu, W.; Niu, Q.; Kong, C. A customizable process planning approach for rotational parts based on multi-level machining features and ontology. *Int. J. Adv. Manuf. Technol.* **2020**, *108*, 647–669. [CrossRef]
- Kang, M.; Kim, G.; Lee, T.; Jung, C.H.; Eum, K.; Park, M.W.; Kim, J.K. Selection and sequencing of machining processes for prismatic parts using process ontology model. *Int. J. Precis. Eng. Manuf.* **2016**, *17*, 387–394. [CrossRef]
- Sugisawa, Y.; Takasugi, K.; Asakawa, N. Machining sequence learning via inverse reinforcement learning. *Precis. Eng.* **2022**, *73*, 477–487. [CrossRef]
- Zhang, C.; Zhou, G.; Hu, J.; Li, J. Deep learning-enabled intelligent process planning for digital twin manufacturing cell. *Knowl. Based Syst.* **2020**, *191*, 105247. [CrossRef]
- Zhao, C.; Dinar, M.; Melkote, S.N. A data-driven framework for learning the capability of manufacturing process sequences. *J. Manuf. Syst.* **2022**, *64*, 68–80. [CrossRef]
- Toaza, B.; Esztergár-Kiss, D. A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems. *Appl. Soft Comput.* **2023**, *148*, 110908. [CrossRef]
- Gong, Y.-J.; Li, J.-J.; Zhou, Y.; Li, Y.; Chung, H.S.-H.; Shi, Y.-H.; Zhang, J. Genetic Learning Particle Swarm Optimization. *IEEE Trans. Cybern.* **2016**, *46*, 2277–2290. [CrossRef] [PubMed]
- Liu, X.; Yi, H.; Ni, Z. Application of ant colony optimization algorithm in process planning optimization. *J. Intell. Manuf.* **2013**, *24*, 1–13. [CrossRef]
- Gómez, J.; Chicaiza, W.D.; Escaño, J.M.; Bordons, C. A renewable energy optimisation approach with production planning for a real industrial process: An application of genetic algorithms. *Renew. Energy* **2023**, *215*, 118933. [CrossRef]
- Jing, X.; Zhu, Y.; Liu, J.; Zhou, H.; Zhao, P.; Liu, X.; Tian, G.; Ye, H.; Li, Q. Intelligent generation method of 3D machining process based on process knowledge. *Int. J. Comput. Integr. Manuf.* **2020**, *33*, 38–61. [CrossRef]
- Sreenivasan, K.S.; Satish Kumar, S.; Katiravan, J. Genetic algorithm based optimization of friction welding process parameters on AA7075-SiC composite. *Eng. Sci. Technol. Int. J.* **2019**, *22*, 1136–1148. [CrossRef]
- Sarvaiya, J.; Singh, D. Selection of the optimal process parameters in friction stir welding/processing using particle swarm optimization algorithm. *Mater. Today Proc.* **2022**, *62*, 896–901. [CrossRef]

25. Peng, H.; Wang, H.; Chen, D. Optimization of remanufacturing process routes oriented toward eco-efficiency. *Front. Mech. Eng.* **2019**, *14*, 422–433. [[CrossRef](#)]
26. Petrović, M.; Vuković, N.; Mitić, M.; Miljković, Z. Integration of process planning and scheduling using chaotic particle swarm optimization algorithm. *Expert Syst. Appl.* **2016**, *64*, 569–588. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.