Journal of Computers Vol. 35 No. 6, December 2024, pp. 169-183
doi: 10.53106/199115992024123506013

Resource Scheduling Strategy and Process Optimization Method for Additive Manufacturing Enterprises Based on Cloud Computing

Liang-Gui Zhang[1,2], Yan-Chao Sun[1*], Wen-Tao Li[1],
Zhi Zhang[1], and Xiao-Bo Dong[1]

[1] Department of Mechanical Engineering, Hebei Institute of Mechanical and Electrical Technology,
Xingtai City 054000, Hebei Province, China
{lianggui3897, yanchao0986, wentao3982, zhang_zhi5278, xiaobo9832}@163.com

[2] Xingtai Metal Processing Equipment Intelligent Integration and Diagnosis Technology Innovation Center,
Xingtai City 054000, Hebei Province, China

Received 22 November 2024; Revised 30 November 2024; Accepted 2 December 2024

Abstract. Cloud manufacturing is a grand manufacturing concept aimed at reducing the waste of manufacturing resources. By borrowing the ideas of cloud computing and utilizing information technology to achieve high sharing of manufacturing resources, the core issues that both the supply and demand sides are concerned about in the cloud manufacturing process are price, product quality, and product production cycle. Therefore, this paper constructs a manufacturing process resource scheduling model that includes three elements. In order to solve the optimal solution of the model, an improved intelligent algorithm is used to solve it. The intelligent algorithm is based on the advantages of the gray wolf algorithm in solving accuracy and convergence speed. The integration of the Bat algorithm in the gray wolf algorithm improves the search ability and performance of the original gray wolf algorithm. At the same time, the gray wolf algorithm can avoid the convergence speed of the Bat algorithm from changing with the increase of iteration times. The phenomenon of slow or almost stopping. To avoid generating local optimal solutions, the accuracy of the cloud resource scheduling model and algorithm solution constructed in this paper were verified through the construction of a simulation environment.

Keywords: cloud manufacturing, scheduling model, grey wolf algorithm, bat algorithm

1 Introduction

Additive manufacturing is one of the rapid prototyping technologies, also known as 3D printing technology. The emergence of 3D printing technology has solved the problem of long product design cycles from design and development to production of finished products. The most significant industry in which additive manufacturing technology reduces production cycles and costs is the injection molding industry, such as the design and production of LEGO bricks, plastic toys, and product casings. The introduction of 3D printing technology can significantly reduce the cost of plastic mold making. Meanwhile, in the field of metal processing, the emergence of 3D printing technology is constantly changing the pattern and production efficiency of the traditional mechanical processing industry.

The foundation of 3D printing technology is mainly based on digital model files, which use adhesive materials such as metal powder during printing. The printing method is layer by layer printing. With the development of technology, traditional manufacturing is gradually transforming towards service-oriented, shared, and intelligent directions. How to integrate more 3D printing enterprise resources and how to more conveniently accept online personalized product customization is the current direction of industry development [1].

In order to cope with the difficulties faced by traditional manufacturing enterprises in terms of timeliness and resource integration, Academician Bohu Li and his team proposed the concept of cloud manufacturing. Cloud manufacturing is a service-oriented, efficient, and knowledge-based networked intelligent manufacturing new model. It extends and transforms existing networked manufacturing and service technologies, virtualizes and servifies various manufacturing resources and capabilities, and conducts unified and centralized intelligent

* Corresponding Author

management and operation. It intelligently perceives, interconnects, collaborates, and processes manufacturing resources throughout the entire manufacturing lifecycle, encapsulates them into networked services, and intelligently completes various activities throughout the product manufacturing lifecycle, with a greater emphasis on assisting physical processing enterprises [2].

Looking at the current situation of traditional manufacturing production and summarizing its existing problems, the following is a summary:

1) The resource types are single and the information updates are slow. In traditional manufacturing systems, there are few resource types and the information update speed is relatively slow. This may lead to low scheduling efficiency and uneven resource allocation during resource scheduling.

2) The degree of automation is low, compared to modern manufacturing systems, traditional manufacturing systems have a lower degree of automation, resulting in more time and resources needed to process tasks in the resource scheduling process, and a higher probability of errors.

3) The information processing capability is limited, and traditional manufacturing systems have obvious deficiencies in information processing and response. The processing methods and response speed of information are slow, which limits the flexibility and accuracy of resource scheduling [3].

The research content of this article is based on the concept of cloud manufacturing, aiming at reducing costs and increasing efficiency for enterprises. How to improve the production efficiency of enterprises and help them obtain better manufacturing resources? Therefore, the composition structure of this article is as follows:

Chapter 2 mainly analyzes the existing research results in cloud manufacturing and resource scheduling optimization. Chapter 3 constructs a resource scheduling model for the three elements of product price, product quality, and completion time that are of common concern to both the supply and demand sides after forming processing tasks. Chapter 4 takes Grey Wolf Algorithm and Bat algorithm as integration objects, draws on the advantages of their respective algorithms, and improves performance in solving accuracy, search speed, and avoiding local optima through algorithm fusion. Chapter 5 is about the simulation process. Through simulation analysis and result comparison, the expected results were obtained, which can then guide actual production.

2 Related Work

The additive manufacturing industry is still in its infancy, with customized printing products and personalized service concepts. Its development is relatively slow and it is difficult to quickly solve the problem of order growth for most 3D printing small and medium-sized enterprises, which hinders their healthy development. In response to the problems existing in the industry, researchers and universities have developed many novel algorithms and methods from different perspectives.

Chaopeng Kuang from Guangzhou University focuses on mold production as his research object. In response to the problems of long production cycles, limited resources, low equipment utilization, lack of communication and cooperation between enterprises in different regions, and insufficient accumulation and inheritance of manufacturing experience and knowledge in traditional mold enterprises, the cloud manufacturing model is introduced to define and classify mold manufacturing resources. A manufacturing resource sharing process model for mold enterprises facing customer orders is constructed, and a mold manufacturing resource sharing cloud service resource library is established to develop an intelligent manufacturing cloud platform system for mold processing [4].

Qiuyun Zhao from Chengdu University of Information Engineering and Technology proposed an adaptive technology framework for manufacturing cloud services, consisting of a data source layer, a data perception layer, a data analysis decision layer, and an action execution layer, to handle sudden abnormal situations in the cloud manufacturing process. In particular, he proposed an adaptive model for manufacturing cloud services driven by a single event and clarified the adaptive process framework [5].

Yingfeng Zhang from Northwestern Polytechnical University proposed a service-oriented packaging and cloud based access method for cloud manufacturing equipment in response to the new demand for service-oriented packaging and cloud based access of manufacturing resources in the cloud manufacturing model. Combining IoT technology, a framework for implementing key technologies such as device side sensor group optimization configuration, manufacturing capability description model, real-time manufacturing service status information active perception, collaborative production and autonomous decision-making of cloud manufacturing equipment, device side manufacturing service packaging, and cloud based access of manufacturing services was designed. This enables the manufacturing capability of processing equipment to be autonomously known, the manufacturing service process information to be transparent and accessible in real time, and can be connected to the cloud manufacturing platform through a loose coupling and plug and play approach, providing a solution for the massive manufacturing resources in cloud manufacturing. Cloud based access, proactive discovery, optimized configuration, And the efficient and high-quality production of production tasks provides a new approach [6].

Chouyong Chen from Hangzhou University of Electronic Science and Technology established a production scheduling model with the optimization objectives of minimizing total weighted completion time, minimizing total processing energy consumption cost, and maximizing machine utilization, taking into account the characteristics of flexible open workshop scheduling in cloud manufacturing environment, and comprehensively considering the collaborative scheduling of cloud tasks and self owned tasks within the enterprise, energy consumption, and utilization of remaining manufacturing resources. Design a dual layer encoding scheme and population evolution strategy for processes and machines based on the multi-objective mayfly algorithm (MMA), and verify the effectiveness of the above model and MMA through simulation experiments [7].

In the field of cloud manufacturing technology, Weijin Jiang from Hunan University of Technology and Business proposes a cloud manufacturing service platform architecture based on a dual chain model to address issues such as difficult trust protection and privacy leakage between transaction parties. The platform stores user data and transaction data on separate chains and uses the Practical Byzantine Fault Tolerance (PBFT) consensus algorithm to solve the data synchronization problem between blocks. At the same time, it proposes to improve the PBFT consensus algorithm by combining the EigenTrust model and Quality of Service (QoS), optimize the selection process and consistency protocol flow of the consensus cluster, and provide steps for rent-seeking and matching manufacturing resources [8].

Yuli Yang from Taiyuan University of Technology proposed a trustworthy evaluation model for cloud manufacturing services that is highly scalable and can better meet personalized needs, addressing the problems of weak scalability and difficulty in meeting personalized needs in traditional models. Firstly, a multi-level and multi granularity trustworthy evaluation framework for cloud manufacturing services was constructed. Based on this framework, a cloud model-based trustworthy evaluation method for cloud manufacturing services was proposed, introducing cloud model theory to uniformly characterize different types of evaluation indicators and describe users' personalized needs. At the same time, the standard deviation method was used to calculate the weight coefficients of different evaluation indicators. The effectiveness of the proposed method was tested through application cases, and the feasibility of the method was verified through comparative experiments of time cost [9].

In the research on resource management of cloud manufacturing, Chen Yang proposed a new manufacturing model based on cloud and software defined network (SDN) - software defined cloud manufacturing (SDCM). This model transferred control logic from automated hardware to software, realizing rapid system reconfiguration, operation and evolution, and adding edge computing. Then, based on the virtualization and flexible network capabilities of SDCM, the data network congestion problem was solved, formalizing the time sensitive data flow control problem for complex manufacturing task sets, and improving the sub task allocation and data routing path. Finally, a method combining genetic algorithm and queuing algorithm was proposed to solve the optimal path [10].

Lingling Yi studied the concepts and structural models of cloud manufacturing and cloud sub chains, the evaluation and construction process of cloud sub chains, and the calculation method of cloud sub chain production capacity in order to facilitate the integration of manufacturing resources and the rational and effective planning and management of available resources in the manufacturing industry. She proposed a method for planning manufacturing resources or service capabilities on cloud manufacturing service platforms from the perspective of sub chains. In addition, from the perspective of multi-directional cooperation among sub chains, she studied the mechanisms for planning negotiation, progress feedback, and conflict resolution between sub chains, and proposed methods for implementing and coordinating cloud manufacturing resource planning at various levels of sub chains [11].

Based on the above research results, this article focuses on the current situation in the additive manufacturing industry, and the work done is as follows:

1) The establishment of a resource scheduling model for cloud manufacturing has been completed, which fully considers the production factors of common concern to both parties involved in manufacturing.

2) For the solution of the model, the Bat algorithm is integrated into the Grey Wolf Algorithm, which improves the performance of the intelligent optimization algorithm in terms of convergence speed, solution accuracy, and convergence speed.

3) We have set up a simulation experimental environment to verify the accuracy of the model and the precision of the optimization algorithm through experiments.

3 Establishment of Manufacturing Task Resource Scheduling Model

Cloud manufacturing is the two-way integration of network and cloud platform operators, managing and configuring resources, providing manufacturing products at low prices and high quality, and delivering manufacturing services to users. By using cloud platforms, resources that were previously limited by geographical location can be interconnected, and real-time information on available cloud resources related to the enterprise can be obtained, thereby achieving resource sharing and reducing various expenses. After receiving the user's task, the cloud platform begins to intelligently search for resources for production based on the user's requirements for task posting [12].

The core problem that resource scheduling faces is the matching problem, which refers to the decision to pair two or more objects. The process of matching 3D printing resource tasks can be seen as the process of satisfying the needs of both parties. This process affects the needs and evaluations of both parties for resources or tasks, thereby affecting the results of demand matching and the satisfaction of participants. Therefore, in the process of establishing a resource scheduling model, the first priority should be to maximize the profits of both parties. The maximization of profits cannot be narrowly regarded as maximizing the profit of a single product, but rather as a comprehensive evaluation that includes various factors in production, considering the core issues of price and product quality that both parties are concerned about, followed by the product production cycle. Therefore, the scheduling model established in this article should include the above three objectives [13].

Meanwhile, the production scheduling optimization model in cloud manufacturing environment needs to be transformed into a mathematical model to describe the production scheduling problem in cloud manufacturing environment and solve the optimal production scheduling scheme through mathematical methods. Therefore, the model in this paper mainly includes the following elements:

1) Production tasks generally include the quantity of production tasks, process requirements, completion time, etc.

2) Production resources: including the quantity, attributes, and utilization efficiency of resources such as equipment, workers, and materials.

3) Task allocation: including assigning production tasks to different production resources, forming a mapping relationship between production tasks and production resources.

4) Production scheduling: including the priority of production tasks, the scheduling order of production resources, and the matching method between production tasks and production resources. Based on the above elements, a mathematical model can be established to describe the production scheduling problem, and the optimal production scheduling plan can be obtained by solving the mathematical model. This model can explain the essence of production scheduling optimization problems in cloud manufacturing environments, quantifying factors such as production tasks, production resources, task allocation, and production scheduling into mathematical models, which can be solved using mathematical methods to achieve the goal of optimizing production scheduling.

3.1 Model Building Process

Firstly, assuming that the set of 3D printing resource providers is A, the representation method of A is:

$$A = \{A_1, A_2, A_3, \cdots, A_n\} \quad (1)$$

Among them, A_i represents the i-th 3D printing resource provider, and assuming that the demand side set in 3D printing is B, the representation method of B is as follows:

$$B = \{B_1, B_2, B_3, \cdots, B_m\} \quad (2)$$

In the formula, B_j represents the j-th 3D printing task demander, and theoretically, the number of demanders is lower than the number of resource providers. In the task allocation process, it is necessary to match the tasks of the task demander and the resource provider. The matching process is shown in Fig. 1.

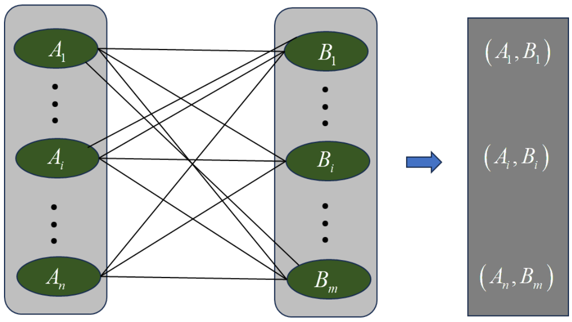

Fig. 1. Schematic diagram of bilateral resource matching process in cloud manufacturing

The revenue matrix of 3D printing resource provider A_i to 3D printing task demander B_j is represented as:

$$\begin{cases} C = \left(c_{ij}\right)_{nm} \\ c_{ij} = \sum_{f=1}^{e} \lambda_f \beta_{ij}^{k} \end{cases} \\ i = 1, 2, \cdots, n \\ j = 1, 2, \cdots, m \\ \lambda_f \in [0,1] \\ \sum_{f=1}^{e} \lambda_f = 1 \tag{3}$$

In the above equation, C represents the revenue matrix of the 3D printing resource provider, and c_{ij} represents the revenue obtained by the 3D printing resource provider A_i during the resource matching process between the 3D printing resource provider A_i and the 3D printing task demander B_j. The revenue matrix of 3D printing task demander B_j for 3D printing task demander A_i is expressed as:

$$\begin{cases} D = \left(d_{ij}\right)_{nm} \\ d_{ij} = \sum_{l=1}^{t} \lambda_l \delta_{ij}^{l} \end{cases} \\ i = 1, 2, \cdots, n \\ j = 1, 2, \cdots, m \\ \lambda_l \in [0,1] \\ \sum_{l=1}^{t} \lambda_l = 1 \tag{4}$$

In the above equation, D represents the revenue matrix of the 3D printing resource provider, and d_{ij} represents the revenue obtained by the 3D printing resource provider B_j during the resource matching process between the 3D printing resource provider A_i and the 3D printing task demander B_j . The 3D printing resource task matching model considering the preferences of both parties is divided into three stages. The first stage is the 3D printing resource task basic attribute constraint model; The second stage is the preference analysis of 3D printing resource tasks; The third stage is the comprehensive benefit model for 3D printing resource tasks. The resources are in the matching process, and the matching flowchart is shown in Fig. 2.

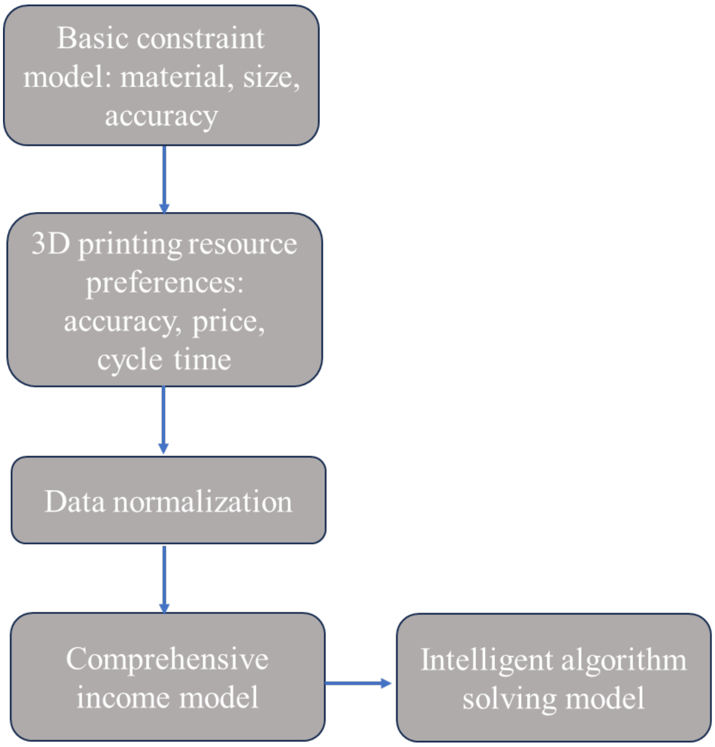

Fig. 2. Process flow chart for matching resource requirements in additive manufacturing process

3.2 Constraints on the Revenue Model of Resource Providers

3D printing processing is similar to traditional machining, and 3D printers must meet basic attribute requirements such as price, material, size, and accuracy of 3D printed parts. Assuming that only one consumable is used in a single printing process, and each consumable is identified by a unique number, the 3D printing resource task matches the constraint formula for materials [14]. When the material M_B required by task demander B_j is the same as the material M_A that resource provider A_i can process, the material constraint result returns a value of 1. When the materials are different, the material constraint result returns a value of 0. The expression is as follows:

$$\varphi_x = \begin{cases} 0 & M_A \neq M_B \\ 1 & M_A = M_B \end{cases} \tag{5}$$

The constraint formula for the size of 3D printing resource task matching pairs, where the length, width, and height of task demand side B_j are represented as l_B , w_B , and h_B, respectively, and must be less than the forming size length L_A, W_A, and H_A that 3D printing resource provider A_i equipment can provide. The above length units are all millimeters, and the expression is:

$$\varphi_y = \begin{cases} 1 & l_B \leq L_A, w_B \leq W_A, h_B \leq H_A \\ 0 & other \end{cases} \tag{6}$$

In the above equation, when the equipment parameters of the resource provider can cover the parameter requirements of the task demander, the return value is 1, and when they cannot cover, the return value is 0. The production accuracy of the product is also a core concern of the task demander, and the product accuracy determines the product quality. Therefore, the product quality is reflected by the leveling processing accuracy. The constraint formula for accuracy matching is as follows:

$$\varphi_z = \begin{cases} 1 & P_A \geq P_B \\ 0 & P_A < P_B \end{cases} \tag{7}$$

Similarly, when the processing accuracy provided by the 3D printing resource provider can meet the processing accuracy required by the 3D printing task requester, the return value is 1, otherwise it is 0. Here, P_A represents the processing accuracy of the resource provider, and P_B represents the processing accuracy of the task requester. Finally, there is the price constraint, which measures whether the budget price of the task demander matches the cost price of the resource provider. The constraint formula is:

$$\varphi_c = \begin{cases} 1 & c_B \leq C_A \\ 0 & other \end{cases} \tag{8}$$

In the formula, c_B represents the budget of the task demander, and C_A represents the cost of the resource provider. After the above constraint description, the overall constraint model for resource matching between 3D task demanders and 3D resource providers on cloud platforms is represented as:

$$\varphi = \varphi_x \cdot \varphi_y \cdot \varphi_z \cdot \varphi_c \tag{9}$$

The accuracy, size, and logistics distance of 3D printing tasks affect the processing difficulty, processing cost, and printing space utilization of 3D printing resources. Studying the accuracy, size, and logistics distance of 3D printing tasks as preferences for 3D printing resources is beneficial for increasing the economic benefits and resource utilization of resource providers. Accuracy is a very important factor in 3D printing processing. The smaller the accuracy value of the workpiece, the smaller the error of the workpiece, but it will increase the processing difficulty and cost of the 3D printer. On the premise of meeting the performance requirements of the processed parts, the resource side should choose higher processing accuracy values as much as possible to make the 3D printer easier to produce and process. In the actual resource matching process, it is rare for 3D resource providers to fully meet the requirements of 3D tasks. Therefore, it is necessary to add corresponding weights to each constraint condition during matching, and consider the matching degree in a single constraint condition, rather than simply matching or not. Therefore, in the process of resource matching on cloud platforms, the following improvements should be made:

$$P_{ij} = \begin{cases} 0 & P_A < P_B \\ 0.85 & P_A \approx P_B \\ 1 & P_A \geq P_B \end{cases} \tag{10}$$

In the formula, when the processing accuracy of the resource provider cannot meet the processing accuracy of the task demander, the accuracy matching preference value P_{ij} is 0. When the processing accuracy of the resource provider is similar to that of the task demander, the preference value is 0.85. In actual setting of cloud platform matching preferences, a range of adjustable accuracy should be provided for similar accuracy. When the processing accuracy of the resource provider fully meets the processing accuracy of the task demand side, the matching preference value is 1. The forming dimensions of different models of 3D printers are different. The 3D printer selects workpieces that are close to its maximum forming size, which can reduce the waste of printing space resources and improve the utilization rate of 3D printer space.

$$Size_{ij} = \frac{l_B \cdot w_B \cdot h_B}{L_A \cdot W_A \cdot H_A} \tag{11}$$

$Size_{ij}$ A represents the preference value of the resource provider's 3D printer for the size of the task provider, where the size preference value is equal to the volume ratio of the task requester to the resource provider's 3D printer. The 3D printing resources and tasks on the cloud manufacturing platform are distributed in different regions. After the processing tasks of the resource side are completed, they need to be delivered to the task side through logistics. Choosing tasks with closer logistics distance can save delivery time and logistics costs. This chapter uses Euclidean distance [15] to calculate the distance between the resource side and the task side.

$$OL_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{12}$$

OL_{ij} represents the Euclidean distance between the task requester and the resource provider, (x_i, y_i) represents the geographic location coordinates of the resource provider, and (x_j, y_j) represents the geographic location coordinates of the task requester. Therefore, according to formula 3, the expression for the revenue model of 3D printing resource providers can be obtained, as follows:

$$c_{ij} = \mu_1 P_{ij} + \mu_2 Size_{ij} + \mu_3 OL_{ij} \tag{13}$$

In the formula, μ_1, μ_2, and μ_3 respectively represent the weight values of the 3D printing resource's accuracy preference, size preference, and logistics distance preference, and the sum is 1.

3.3 Task Demand Side Benefit Model

The 3D printing task party mainly focuses on the price and delivery time of the product, and the unit price is mainly composed of material costs, labor service fees, machine usage fees, and logistics costs. Finally, the pricing will be based on the weight of the model provided by the 3D printing task party. The total delivery time of the order includes processing time and logistics time. Therefore, the benefit function of the task demand side is represented by formula 22 as follows:

$$d_{ij} = \mu_4 Q_{ij} + \mu_5 T \tag{14}$$

Where Q_{ij} represents the quotation of the 3D printing task resource provider, and T represents the total time required for processing and logistics. μ_4 and μ_5 represents the weight allocation of the task demander in terms of price and delivery time, and the sum is 1. The revenue models of the task demander and resource provider are obtained through formulas 21 and 22, and are therefore expressed as follows:

$$e_{ij} = \mu_1 P_{ij} + \mu_2 Size_{ij} + \mu_3 OL_{ij} + \mu_4 Q_{ij} + \mu_5 T \tag{15}$$

After the establishment of the above model and the creation of constraint models, this article has provided a model foundation for maximizing profits. In the cloud manufacturing environment, a 3D printing resource task matching model based on the preferences of both parties has been completed. Firstly, the preference information of the resource provider and the task provider was analyzed, and then a basic attribute constraint model, preference benefit function, and comprehensive benefit model were constructed. The results showed that the preferences of 3D printing resources and 3D printing tasks emphasized complementary relationships, and unilateral decision-making would hinder the generation of pairs. Adding stable constraints to the matching scheme can obtain a stable set of matching pairs, proving the feasibility of considering the preferences of both parties in the 3D printing resource task matching model. Furthermore, the objective model of this paper was solved through intelligent algorithms.

4 Establishment of Manufacturing Task Resource Scheduling Model

The task matching environment for cloud manufacturing 3D printing resources typically involves high-dimensional data and multiple variables. The Grey Wolf Algorithm has excellent problem-solving ability when dealing with high-dimensional and large-scale data, and the model can make more accurate predictions and decisions. The Grey Wolf Algorithm can achieve end-to-end learning, that is, learning the final decision from raw data. This learning method is more flexible and intelligent than rule-based methods. At the same time, the algorithm can also learn different decision-making strategies based on the needs and preferences of different users, thereby achieving personalized service combinations. This chapter proposes a 3D printing resource task matching model based on Grey Wolf Algorithm, considering the complex, dynamic, and multi task scenarios of 3D printing resource task matching in cloud manufacturing environment.

In the process of solving the model solution, intelligent algorithms need to be used to optimize the solution. Through analysis, this paper chooses the Grey Wolf Algorithm as the optimization algorithm, while considering the shortcomings of grey wolf in optimization solving, and improves the Grey Wolf Algorithm.

4.1 Algorithm Improvement Plan

The Grey Wolf Optimizer (GWO) is a relatively new swarm intelligence optimization algorithm developed by researchers Mirjalili [16]. Its main idea is to optimize related searches by leveraging the natural hunting behavior of grey wolf populations. The Grey Wolf Algorithm can demonstrate excellent performance in terms of solving accuracy and convergence speed for related problems.

However, the Grey Wolf Algorithm itself also has some flaws, which can lead to incorrect results in the process of resource scheduling. Therefore, it is necessary to analyze and summarize the shortcomings of the Grey Wolf Algorithm, and make targeted improvements based on its flaws. The main flaws of the Grey Wolf Algorithm include:

1) Poor population diversity: The initial population of the Grey Wolf Algorithm is generated through random initialization, which cannot guarantee good population diversity and may cause the algorithm to fall into local optima during the search process.

2) Slow convergence speed in the later stage: During the search process of the Grey Wolf Algorithm, the wolf pack mainly judges the distance to the prey based on the distance to alpha, beta, and delta, resulting in a slow convergence speed in the later stage.

3) Easy to fall into local optima: Due to the fact that alpha wolves may not necessarily be the global optimum, in continuous iterations, ω constantly approaches the top three wolves, which may cause the Grey Wolf Algorithm to fall into local optima.

Easy premature convergence: The Grey Wolf Algorithm may experience premature convergence when facing complex problems, resulting in low convergence accuracy and slow convergence speed. But the search ability is not strong, so the Bat algorithm is integrated into the Grey Wolf Algorithm [17].

The Bat algorithm is an emerging swarm intelligence optimization algorithm proposed by Yang, a scholar at the University of Cambridge, based on the use of echolocation technology by bats in nature to perceive their surrounding environment and locate food or obstacles. The optimization process of the Bat algorithm is completed through iterative techniques. In the initial iteration, the positions of individuals in the bat population are randomly initialized as candidate solutions, and then continuously explored and updated through subsequent iterations until the ideal solution is found. The entire process of the Bat algorithm searching for the optimal solution is not static and unchanging, but constantly evolving. From the initial state of population search being unordered and random, the state of population search gradually changes to an ordered search for the global optimal solution through subsequent iterative search processes. The basic unit of Bat algorithm optimization is each bat in the bat population, and each bat individual has a corresponding fitness value. When bat individuals perform optimization in the solution space of the problem, they will make corresponding changes based on their fitness value. When bats hunt targets and identify prey, they use tuning techniques to regulate the dynamic behavior of bat populations, and by adjusting the parameters in the algorithm, they achieve the optimization of the algorithm for problem solving.

According to the principle of the Bat algorithm model, in the early stage of the optimization process, the convergence speed and global search ability of the Bat algorithm are very strong. However, as the number of iterations increases, the convergence speed in the later stage will slow down or even almost stop, resulting in the algorithm falling into local optimal solutions.

4.2 Algorithm Solving Process

After algorithm improvement and taking the advantages of each algorithm, the improved Grey Wolf Bat algorithm is obtained [18]. The solving steps of the improved algorithm are as follows:

Step 1: Initialize the grey wolf population, such as initializing the positions of individuals in the population, and randomly initialize them based on the upper and lower limits of their positions;

Step 2: Initialize parameters, including parameters from the Grey Wolf Optimization Algorithm and various parameters from the Bat algorithm;

Step 3: Calculate the fitness value of each individual in the gray wolf population based on the existing fitness function, and save the three individuals with the best fitness values.

Step 4: Update the positions of individuals in the population.

Step 5: After updating the location, update the parameters in the algorithm and repeat step 3;

Step 6: Initialize the bat population and update the first three bat individuals based on the three individuals with the best fitness values obtained in Step 5. Randomly initialize the remaining bat individuals based on the upper and lower limits of their positions. Calculate the specific fitness value of each bat based on the existing fitness function, and obtain the optimal value, which is recorded as the current optimal solution of the bat;

Step 7: Optimize the speed of individual bats and update their positions.

Step 8: Generate a random number. When the random number passes the set value, select a solution from the optimal solution set and randomly fly near the selected optimal solution to generate a local new solution. When the random number is lower than the set value, a local new solution is generated near itself.

Step 9: Sort all bats according to their fitness values and select the global optimal solution;

Step 10: Determine whether the termination condition is met. If the termination condition is not met, return to step 3 to continue searching. If it is met, output the global optimum.

The algorithm solving flowchart is shown in Fig. 3.

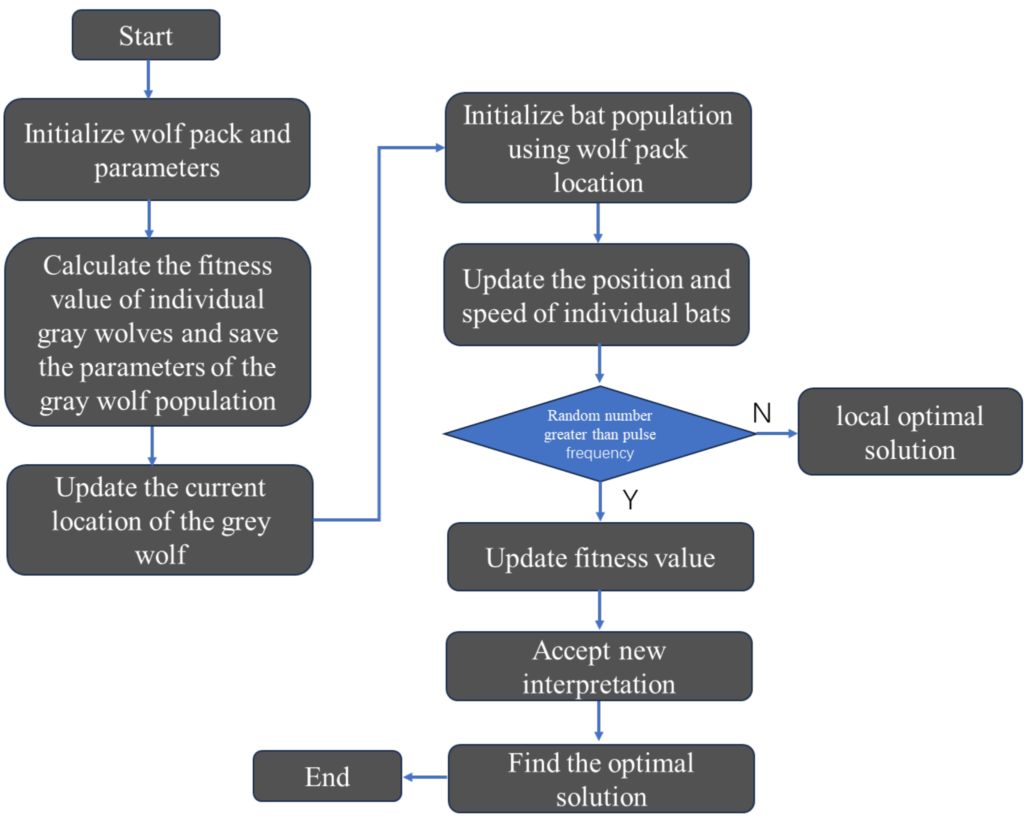

Fig. 3. Flow chart of cloud manufacturing resource scheduling with improved algorithm

The pseudocode for improving the algorithm is as follows:

```
Pseudo code
# Initialize parameters
Initialize parameters (population size, number of iterations, etc.)
Initialize the position of grey wolves (alpha, beta, delta, and omega wolves)
Initialize the position and velocity of bats
# Main loop
for iteration = 1 to MaxIterations do
  # Grey Wolf Optimizer part
  # Evaluate fitness of all individuals (wolves and bats)
for each individual in population do
# Bat Algorithm part
    # Update the velocities and positions of bats
# Combine the best solutions from GWO and BA
  # Optionally, you can use a weighted average or other merging strategy
  Best_solutions = MergeBestSolutions(GWO_best_solutions, A_best_solutions)
    # Update the alpha, beta, delta positions using the combined best solutions
  UpdateAlphaBetaDelta(Best_solutions)
    # Log current iteration information
  LogIterationInfo(iteration, Best_solutions)
end for
 # Return the best solution found
Return Best_solution
```

By improving the basic grey wolf algorithm and incorporating the bat algorithm, the improved algorithm can overcome their respective shortcomings and provide an algorithm flow. Through the improved algorithm, the optimal solution for resource scheduling is obtained, providing an algorithm solution for further simulation solutions. In the next chapter, the accuracy of the model and the effectiveness of the algorithm constructed in this paper will be verified through real cloud resource scheduling cases.

5 Establishment of Manufacturing Task Resource Scheduling Model

This section completes the process design for implementing cloud resource scheduling on the CloudSim [19] simulation platform. The simulation process of resource allocation in CloudSim is shown in Fig. 4:

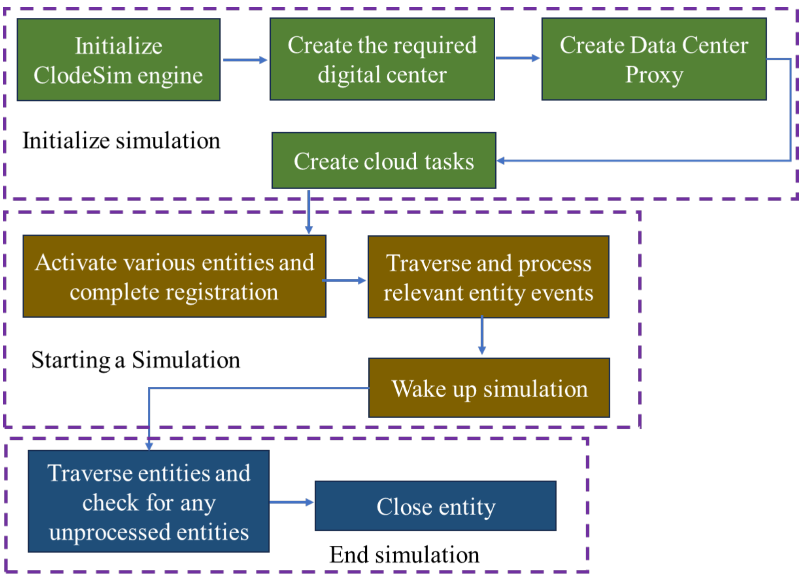

Fig. 4. Simulation flowchart of resource allocation

The experimental data settings and hardware parameters are shown in Table 1.

Table 1. Computer simulation environment and algorithm parameter list

Parameter	Parameter values
CPU	Core i7
Operating system	Windows 10
Memory capacity	8GB
Hard disk	1T
Pulse frequency range	[−3, 3]
Maximum pulse emission frequency	0.7
Maximum audio intensity	0.65
The size of bat populations	20
The size of wolf pack populations	50
Maximum Number of Iterations	1000
Algorithm dimension	30

The software environment for simulation experiments is mainly completed through the VScode programming environment and Matlab. The algorithm process is written using C++instructions, and the simulated data is analyzed using Matlab.

The convergence curve of an algorithm can directly reflect its convergence speed, ability to escape local optima, and solution accuracy, and is an important indicator for measuring algorithm performance. In order to compare the advantages of the algorithm proposed in this article, Bat algorithm, Grey Wolf Algorithm, and improved fusion algorithm are used for comparison. For ease of description, the improved grey wolf Bat algorithm is referred to as I-GWO-BA, and the three improved algorithms are run on the Schwefel function for convergence. The convergence result of the algorithm is shown in Fig. 5.

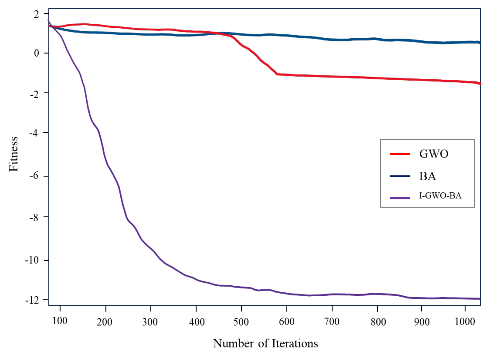

Fig. 5. Convergence results of three comparative algorithms

From the graph, it can be seen that in the first 400 iterations, both BA and GWO algorithms have slow convergence speeds, while BA's accuracy in solving is higher than GWO's. However, after 450 iterations, BA's convergence speed accelerates, and the solving accuracy gradually surpasses BA. Then, at 600 iterations, the convergence efficiency deteriorates and falls into local constraints. Throughout the entire iteration process, the algorithm in this chapter remained in a convergent state compared to the other two algorithms, with the highest solution accuracy.

5.1 Simulation Analysis of Resource Scheduling Results in Cloud Manufacturing Environment

This experiment evaluates the performance differences of algorithms when the number of tasks changes. When conducting experimental evaluations, compare the algorithm proposed in this paper with the Grey Wolf Algorithm and Bat algorithm. During the experiment, keep the parameters of the host, virtual machine entity, and scheduling algorithm consistent, set the task quantity hierarchy to (3000, 4500, 6000, 7500, 8500), and calculate the time span for executing tasks at different levels. The experimental results are shown in Fig. 6.

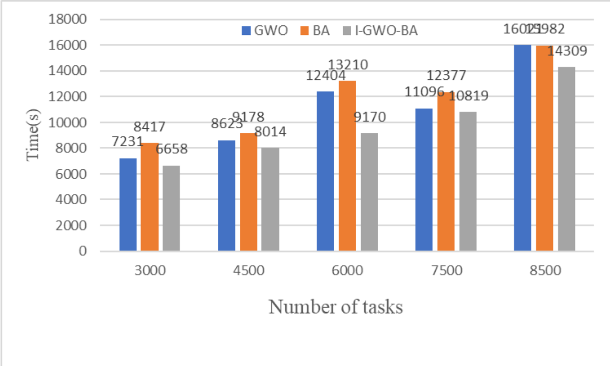

Fig. 6. Comparison chart of task completion time

The experimental results can lead to the following conclusions:

1) As the number of tasks increases, the completion time of all three scheduling algorithms is also increasing;

2) When the number of tasks is the same, the BA algorithm has the highest average completion time, while the difference in completion time between the GWO algorithm and the BA algorithm is subtle. Sometimes the GWO algorithm has a shorter time, and sometimes the BA algorithm has a shorter time. However, overall, the completion time of the I-GWO-BA algorithm is lower than that of the GWO algorithm and the BA algorithm. Due to the creation of time constraint functions in the improved algorithms proposed in this article, the I-GWO-BA algorithm is significantly better than the original algorithm in terms of completion time. Based on the number of tasks set in this article, the algorithm has reduced the overall task completion time by 14.39% compared to the original algorithm.

In actual cloud computing services, users rent cloud resources according to their needs, and one leasing method is "pay as you go". Each virtual machine has the attribute of cost per unit time. Therefore, the total cost can be calculated based on the execution time of each user task on the virtual machine and the cost per unit time. The purpose of this experiment is to verify whether the cost of using the I-GWO-BA algorithm for cloud resource scheduling has been reduced. During the experiment, the execution costs of different task quantities were calculated while maintaining consistent configurations for each entity.

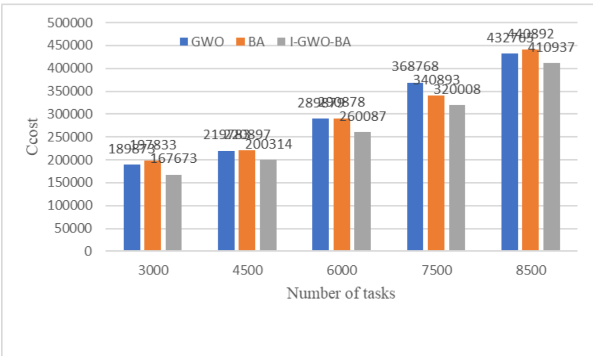

Fig. 7. Comparison chart of algorithm profit results

The following conclusions can be drawn from the results of Fig. 7:

1) As the number of tasks increases, the benefits generated are also increasing;

2) When the number of tasks is the same, the I-GWO-BA algorithm has the highest benefit. Compared with the GWO algorithm, the benefit of the I-GWO-BA algorithm has increased by 10.45%, and compared with the BA algorithm, its benefit has increased by 9.74%. The superiority of the I-GWO-BA algorithm in resource scheduling is mainly attributed to the addition of a cost constraint function in the algorithm, which takes into account the product prices that are of concern to both parties in cloud manufacturing. When redefining pheromones using the processing cycle cost function, enterprises with short production cycles, high processing accuracy, and low single product prices are more likely to be selected. Therefore, this algorithm effectively improves the process benefit of cloud manufacturing tasks.

6 Conclusion

This article studies and analyzes the development process of cloud manufacturing, the current research status at home and abroad, resource scheduling algorithms in the environment, and the application of traditional Grey Wolf Algorithm in cloud computing environment through reading relevant literature and research results. In response to some problems in the Grey Wolf Algorithm, a comprehensive and improved Grey Wolf Algorithm is proposed. The specific improvement operations include using the Bat algorithm to initialize the population during population initialization, improving the step size formula in the local search operation of the Grey Wolf Algorithm, replacing the local optimal value, introducing weights to each individual wolf pack, and updating the global optimal strategy, thereby improving the performance of the algorithm, shortening the time for resource scheduling in cloud computing environment, balancing the load of various resources, increasing the profits of both manufacturing parties, and proving the feasibility and effectiveness of the improved algorithm in resource scheduling application in cloud computing environment through simulation experiments on CloudSim simulation platform.

With the continuous development and exploration of the Internet of Things by intelligent humans, this cloud platform, which is mainly used to collect and store massive amounts of data, will undoubtedly provide higher quality and more efficient services for human production and life through big data and artificial intelligence. The current hot topics of cloud computing, artificial intelligence, the Internet of Things, and big data are closely related to each other. They influence each other and jointly promote the arrival of the Fourth Industrial Revolution of mankind. In this macro context, based on the advantages of the grey wolf and Bat algorithms, this article proposes an improvement plan for the hybrid grey wolf and Bat algorithm by combining it with other related algorithms. The conclusion drawn from the experiment can strongly prove that the comprehensively improved Grey Wolf Bat algorithm has achieved significant improvements in convergence speed, accuracy, and optimization ability. Due to limitations in research level, the research direction for this step is:

When constructing the profit model, factors such as carbon emissions and energy consumption were not fully included, so the calculation and quantification of carbon emissions and energy consumption in the process of processing enterprises were taken as the research focus. In the process of considering resource load, this article only involves two aspects: the central processing unit (CPU) and memory, and does not include bandwidth and other aspects in the scope of consideration. 3. The optimal resource matching obtained in this article only matches one enterprise, but the production process of a product involves multiple process flows. Under the premise of maximizing profits, breaking down the process and achieving scheduling for each process can further refine the scheduling.

References

[1] L. Chen, J.-K. Zhang, M. Cao, B.-Z. Mei, S.-J. Qi, C.-Y. Xie, Y.-M. Deng, Overview of advanced additive manufacturing technology in die and mold industry, Journal of Ningbo University(Natural Science & Engineering Edition) 36(6) (2023) 1-16.

[2] M.-H. Yuan, Z.-C. Li, H.-Y. Huang, F.-Q. Pei, H.-Y. Yu, Research on clustering method of resource service cluster in cloud manufacturing, Manufacturing Technology & Machine Tool (10)(2023) 41-47. DOI: 10.19287/j.mtmt.1005-2402.2023.10.006.

[3] H.-W. Wang, Y.-F. Zhang, G.-Z. Peng, K.-H. Yan, Review of cloud manufacturing based on resource flow, Journal of Huazhong University of Science and Technology (Natural Science Edition) 50(6)(2022) 11-30.

[4] C.-P. Kuang, J.-H. Tao, T.-T. Li, B. Chen, Y. Ma, W. Wang, Research on resource sharing technology and platform development of mould manufacturing in cloud manufacturing environment, Manufacturing Technology & Machine Tool (2)(2023) 90-96.

[5] Q.-Y. Zhao, L. Wei, H.-P. Shu, Adaptive Technology Framework of Manufacturing Cloud Service in Cloud Manufacturing Environment, Journal of Chengdu University of Information Technology 36(1)(2021) 45-50.

[6] Y.-F. Zhang, G. Zhang, T. Yang, J.-Q. Wang, S.-D. Sun, Service encapsulation and virtualization access method for cloud manufacturing machine, Computer Integrated Manufacturing Systems 20(8)(2014) 2029-2037.

[7] C.-Y. Chen, Z. Zhang, H.-J. Li, Z.-L. Zhang, Multi-objective Mayfly Algorithm for Flexible Open Shop Scheduling Problem in Cloud Manufacturing Environment, Journal of Information and Management 8(6)(2023) 1-17.

[8] W.-J. Jiang, W.-Y. Zhou, E. Li, T.-T. Luo, Y. Yang, Research on cloud manufacturing service architecture and consensus algorithm based on blockchain technology, Chinese Journal on Internet of Things 7(1)(2023) 159-173.

[9] Y.-L. Yang, Y.-H. Li, A.-H. Deng, Trust Evaluation Model of Cloud Manufacturing Services for Personalized Needs, Computer Science 49(3)(2022) 354-359.

[10] C. Yang, F.-Y. Liao, S.-L. Lan, L.-H. Wang, W.-M. Shen, G.-Q. Huang, Flexible Resource Scheduling for Software-Defined Cloud Manufacturing with Edge Computing, Engineering 22(3)(2023) 60-70.

[11] L.-L. Yi, J.-F. Tu, X.-X. Liang, Cloud manufacturing resource planning and its coordination mechanism based on supply sub-chain, Journal of Mechanical & Electrical Engineering 33(12)(2016) 1442-1447.

[12] X.-J. Yang, B. Yang, F. Wu, Research on the optimization of cloud manufacturing resource configuration based on the improved adaptive cuckoo algorithm, Manufacturing Technology & Machine Tool (5)(2018) 137-142.

[13] H.-B. Su, Cloud source scheduling strategy based on improved Grey Wolf Optimization Algorithm, Intelligent Computer and Applications 14(2)(2024) 28-34.

[14] Y.-K. Wang, S.-L. Wang, B. Yang, S.-B. Wang, Dynamic Adaptive Reconfiguration Method for Cloud Manufacturing Service Composition in Practical Multi-constraint Environment, Journal of Mechanical Engineering 59(14)(2023) 339-351.

[15] F.-S. Yu, Y.-J. Wang, X.-H. Xu, Study of Operational Associated Rules of Air-conditioning Water System under Unconventional Conditions Based on Minimization of Euclidean Distance, Refrigeration & Air Conditioning 37(3)(2023) 337-343.

[16] C.-G. Chang, J.-X. Chen, Research on Production Scheduling Optimization of Prefabricated Components Based on Grey Wolf Optimization Algorithm, Journal of Shenyang Jianzhu University (Social Science) 25(6)(2023) 589-596.

[17] W.-J. Cheng, S.-F. Lin, L.-F. Xu, C.-T. Liu, D.-D. Li, Y. Fu, System-side harmonic impedance estimation method based on minimum impedance deviation criterion and improved adaptive bat algorithm, Electric Power Automation Equipment 42(11)(2022) 183-189.

[18] H.-C. Yang, Y.-B. Li, Flexible Assembly Shop Green Scheduling Based on Improved Grey Wolf Optimizer Algorithm, Digital Manufacture Science 21(2)(2023) 151-156.

[19] H.-Q. Zhang, X.-P. Zhang, H.-T. Wang, Y.-H. Liu, Task Scheduling Algorithm Based on Load Balancing Ant Colony Optimization in Cloud Computing, Microelectronics & Computer 32(5)(2015) 31-35+40.