

Article

# Matching of Manufacturing Resources in Cloud Manufacturing Environment

Yanjuan Hu \*, Leiting Pan, Dongwei Gu, Zhanli Wang, Hongliang Liu and Yilin Wang

School of Mechatronic Engineering, Changchun University of Technology, Changchun 130012, China; 2202101022@stu.ccut.edu.cn (L.P.); gudongwei@ccut.edu.cn (D.G.); wangzl@ccut.edu.cn (Z.W.); liuhongliang@ccut.edu.cn (H.L.); wangyilin@ccut.edu.cn (Y.W.)

\* Correspondence: huyanjuan@ccut.edu.cn

**Abstract:** With the introduction and application of new information technology in manufacturing, various advanced manufacturing models and national strategies have received more and more attention. The goal of cloud manufacturing is to closely link the resources and capabilities of manufacturers through a variety of services to create a dedicated platform for complex manufacturing process needs. How to achieve effective matching of various manufacturing resources and capabilities in the form of services will be a common problem in the future. In order to effectively improve cloud manufacturing tasks and resource matching efficiency and save resources, this study considers the common aspects of cloud manufacturing resource matching as service quality indicators, and extends the scope to the requirements of manufacturing resources, and the matching pattern of traditional service resources. There are additional restrictions on the resource service matching process. At the same time, the resource service matching is usually asymmetric. Therefore, we introduce the concept of task complexity of demand resources, and propose a combination system based on task complexity and service quality evaluation. The artificial bee colony algorithm (ABC) is used for analysis and verification. The experimental paper further validates the proposed the feasibility and efficiency of the method.

**Keywords:** cloud manufacturing; manufacturing resources matching; artificial bee colony algorithm (ABC); task complexity; analytic hierarchy process (AHP); asymmetric



**Citation:** Hu, Y.; Pan, L.; Gu, D.; Wang, Z.; Liu, H.; Wang, Y. Matching of Manufacturing Resources in Cloud Manufacturing Environment. *Symmetry* **2021**, *13*, 1970. <https://doi.org/10.3390/sym13101970>

Academic Editors: Jan Awrejcewicz and Tomohiro Inagaki

Received: 16 August 2021  
Accepted: 9 October 2021  
Published: 19 October 2021

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



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

## 1. Introduction

With the development of emerging manufacturing technologies and Internet technologies, production has gradually developed into multi-role and multi-field collaborative design and production, and there are many cross-regional and even cross-domain resources and technologies combined with production. However, due to the uneven distribution of manufacturing resources, production is faced with a certain contradiction: many small and medium-sized enterprises are faced with the lack of talents and technology, processing equipment is limited and backward, and some enterprises and colleges have long-term idle hardware resources, human resources and software resources. In the long run, this will not only waste social resources and affect the innovation and competitiveness of enterprises, but will also restrict the rapid development of social productivity. The existing manufacturing models, such as application service provider (ASP) [1] and manufacturing grid [2], can no longer meet the production requirements well. In this context, the concept of “cloud manufacturing” was proposed and discussed [3,4], aimed at solving the problems of poor manufacturing innovation, backward manufacturing models, low resource utilization rate, decentralized manufacturing resources and regionalization, realizing the optimal matching of manufacturing resources, and improving the independent innovation and market competitiveness of enterprises. Subsequently, many researchers participated in researching the cloud manufacturing platform. For example, in order to solve the problem of centralized use of heterogeneous resources and business units distributed in different subsidiaries, a

new hybrid private cloud framework was proposed [5], which provided an important reference for the establishment of cloud manufacturing platform resource-matching framework. Starting from the traditional cloud platform scheduling rules, a new task scheduling agent model was proposed [6], which solved the short-term resource scheduling problem in the small-scale and high-precision environment of the cloud platform. Facing the complex manufacturing and scale collaborative manufacturing environment, and considering the concept of cloud manufacturing and its operating principle, a manufacturing cloud service capability model for describing machine tool services was proposed [7], which solved the problem of manufacturing resource matching. At the same time, on the basis of considering multi-objectives and statistical characteristics, a super network solution and cloud platform framework with related functions were proposed [8], which solved the dynamic supply and demand matching and scheduling problems of manufacturing resource services. With the in-depth study of the cloud platform resource-matching framework, the resource allocation [9] and task scheduling problem in the cloud platform environment have become more and more obvious. Numerous researchers have found that cloud platform on-demand matching and resource availability have become ideal for scientific workflow applications. The application can start with a minimum number of resources and allocate more resources when needed; therefore, resource allocation and task flow scheduling problems based on heuristic algorithms were proposed. For example, through the establishment of an integrated computing resource allocation model, utilizing an improved niche immune algorithm was proposed [10], which solved the problem of optimal allocation of computing resources. The experiment proved the validity of the designed heuristic information, and showcased the high performance of NIA to solve the optimal allocation of computing resources compared to other intelligent algorithms. At the same time, in the realization of the fast and effective matching of supply and demand of cloud manufacturing resources, a method for studying the diversity of matching processes between intelligent matching engine and cloud was proposed [11]. By establishing a car/motorcycle attachment ontology database, combining quantitative methods, matching algorithms and semantic similarity, effective matching of cloud manufacturing resources is realized. On this basis, in order to further study the problem of rapid and effective allocation of resources in the private cloud of manufacturing enterprises, a sequential resource allocation bus architecture combined with genetic algorithm was proposed [12], which realized the rapid and effective allocation of manufacturing enterprise private cloud resources. Considering a large number of homogeneous resources and dynamic customer demand constraints, including the problem of how to measure fuzzy QoS and choose the best service considering design preferences, a service quality model based on cloud manufacturing platform design preferences was proposed [13] and through particle swarm optimization algorithm (PSO), helped customers get the best manufacturing service. Focusing on the development of crowdsourcing products and how to use a large number of service resource combinations in a cross-platform way in the cloud crowdsourcing environment, the resource matching and combination optimization model of cloud service product development was established [14], and cooperative bacterial foraging optimization (CBFO) was used to obtain a resource combination.

Summarizing the current research on the matching of cloud manufacturing resources, the main finding is that a lot of research has been done on the allocation architecture and resource combination mechanism, and the cloud manufacturing resource matching mechanism has been improved and developed to a large extent. However, in view of the diversity and complexity of cloud manufacturing tasks, there is a lack of corresponding cloud manufacturing resource matching models and methods. Considering the above factors, based on the research and in reference to the existing research results, this article focuses on the problems of resource matching in the cloud platform system.

## 2. Cloud Manufacturing Resource Classification

Manufacturing resources are the basis of resource management and resource matching. Therefore, the resources should be classified scientifically and rationally before match-

ing [15], which will affect the rationality of resource matching and affect the user experience. With the advent of cloud computing and the era of big data, manufacturing resources in a broad sense have already had new connotations [16], they have witnessed new developments compared to manufacturing resources in traditional manufacturing models. Manufacturing resources play a decisive role in resource location, packaging, matching, virtualization construction, and information feedback on cloud manufacturing platforms. There is no uniform standard for the classification of resources due to the different purposes of the service, but there is still a commonality in essence. Manufacturing resources consist of enterprises, equipment, workshops, software, manpower, technology, and other factors, and there are individual differences between them [17]. Hard resources do not participate in other tasks until they match the specified task and then end their service. Hard resources include: material resources, processing equipment resources, and human resources. When a soft resource provides services to users, it is not restricted by geography, time, processing, etc., and can serve multiple tasks while matching services to specified tasks. Soft sources include: knowledge resources, user information resources, data resources, and software resources.

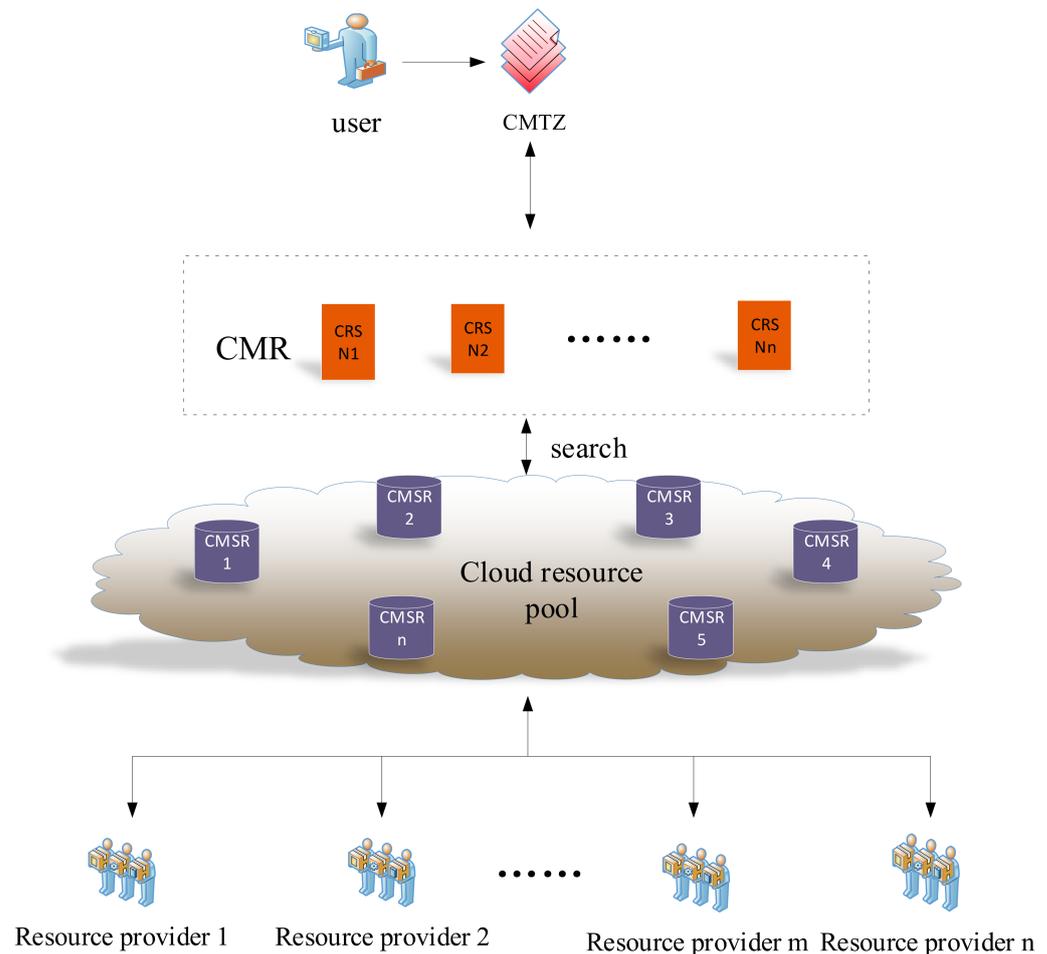
### 3. Matching Problem of Cloud Manufacturing Resources

The problem of matching cloud manufacturing service resources [18] is that the users enter the platform search and compare the manufacturing resource requirement ontology with the description body of the existing manufacturing resources of the cloud platform, so as to find the manufacturing resources that meet the requirements. The cloud manufacturing task and resource-matching process is also an iterative process. First, through the basic attributes (resource name, type, specifications, etc.) of the existing resources of the cloud platform, status attributes (idle, received orders, processing, maintenance, stop use, scrap), functional attributes (manufacturing process, manufacturing accuracy, manufacturing parameters, and other information) and task requirements information constraints (basic information constraints, functional information constraints, state information constraints match), a set of candidate resources sets will be established [19]. Second, through the information constraint of manufacturing resources (time T, quality Q, cost C, reliability Re, recoverability G), the service resources satisfying the user requirements are searched; that is, the optimal resource service chain is formed.

#### 3.1. Matching of Resources and Manufacturing Task Requiring a Single Resource

The matching process of tasks and resources [20,21] is the key link in cloud manufacturing, so the problem of efficiently and accurately matching manufacturing resources for manufacturing tasks is the research focus of this paper. The task complexity of the demand resource is judged by the information constraint of the task requirement. There are two kinds of task complexity of demand resources: the first requires the manufacturing task of a single resource, and the second requires the manufacturing task of diverse resources. When a single resource manufacturing task enters the cloud platform, the cloud platform selects a plurality of manufacturing resources that meet the user requirements from the mass manufacturing resources through task attribute information constraints and various attribute analysis of the manufacturing resources to form a candidate resource set. In the candidate resource set, the user's constraints on manufacturing resources further narrows the candidate resources, and the artificial bee colony algorithm [22,23] solves the model to obtain the optimal manufacturing resources to complete the manufacturing task. When a user requests a cloud platform to create a single resource, the cloud platform analyzes the requirements of the manufacturing resource and the attributes of the existing resources of the platform, a plurality of manufacturing resources satisfying the processing requirements are searched from the cloud platform resource pool to form a candidate resource set. Entering the matching link between the cloud manufacturing task and the resource, the manufacturing task requiring the single resource is matched with the candidate service resource that satisfies the user's constraint on the manufacturing resource in the candidate

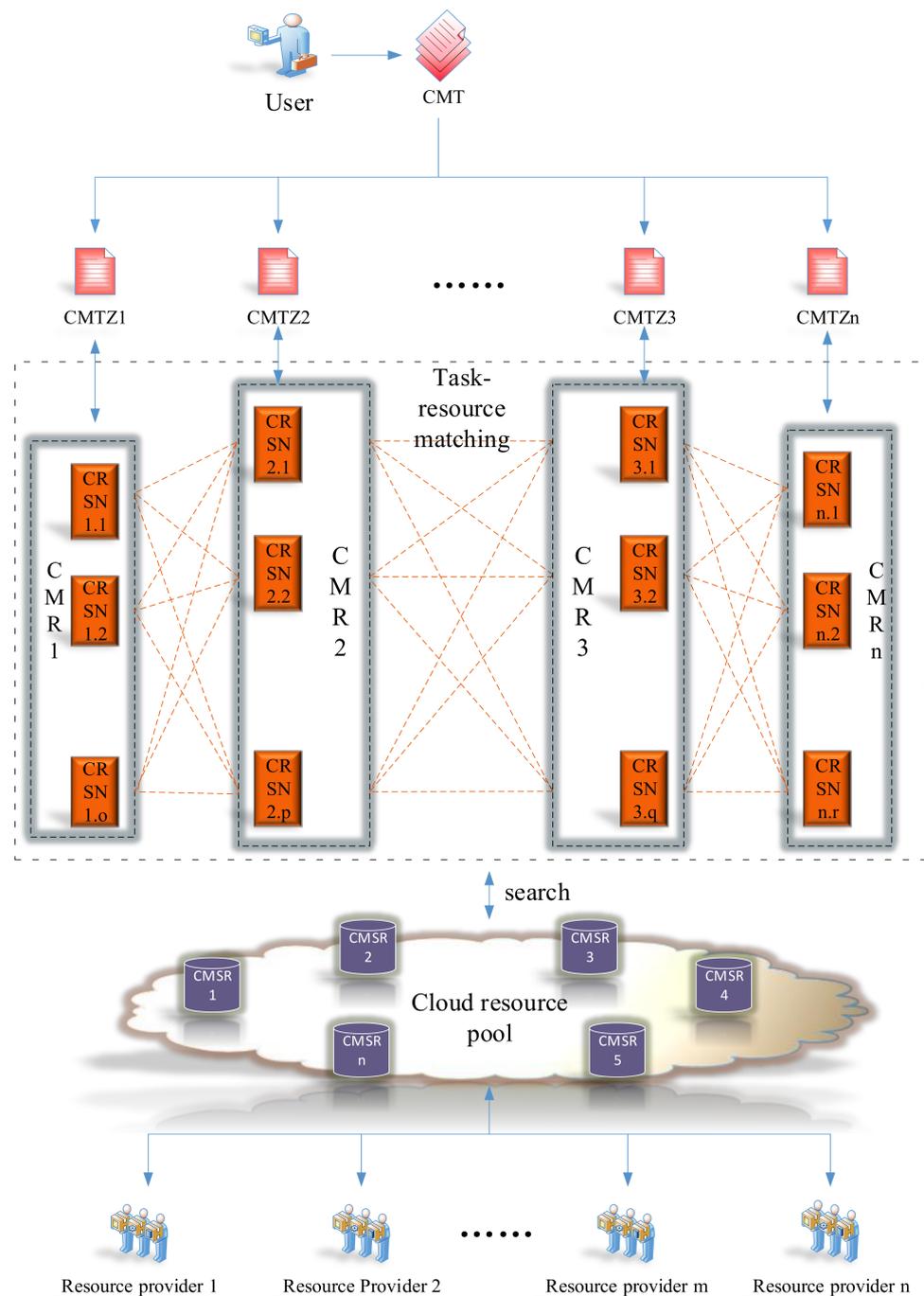
resource set, and finds the resource that meets the requirements to complete the task and the resource matching. Figure 1 shows the resource manufacturing task matching process that requires a single resource.



**Figure 1.** Matching process diagram of the single resource is needed. *CMTZ* represents the manufacturing task that requires a single resource; *CMR* represents a candidate resource set; *CRSN<sub>i</sub>* represents the *i*th candidate service resource; *CMSR<sub>i</sub>* represents the resources of the *i*th cloud platform.

### 3.2. Matching of Resources and Manufacturing Task Requiring Multiple Resources

Manufacturing tasks that require multiple resources can be seen as a combination of manufacturing tasks that require a single resource. When a user proposes a manufacturing task that requires diverse resources to the cloud platform, the cloud platform will select a plurality of service resources that meet the manufacturing task requirements in accordance with the requirements of the user for the manufacturing resources and the various attributes of the manufacturing resources from the resource pool, and form a plurality of one-to-one mapping relationship between the manufacturing task requiring a single resource and the plurality of candidate resource sets. The matching process finds the optimal solution in the corresponding candidate resource set according to the user's constraints on the manufacturing resources. The next manufacturing task is bound by the user's requirements for manufacturing resources and the various attributes of the manufacturing resources and the optimal resources searched by the previous manufacturing task (transportation costs and time constraints between subtasks and subtasks). Finally, each set of candidate resources corresponding to the manufacturing task requiring a single resource will select a manufacturing resource that best meets the processing requirements, and jointly fulfill the manufacturing task requiring multiple resources. Figure 2 shows the matching process of resources and manufacturing task requiring multiple resources.



**Figure 2.** Matching process of the multiple resources are needed. *CMT* represents the manufacturing task requiring multiple resources;  $CMTZ_i$  represents the  $i$ th manufacturing task requiring a single resource;  $CMR_i$  represents the  $i$ th candidate resource set;  $CRSN_i$  represents the  $i$ th candidate service resource;  $CMSR_i$  represents the resources of the  $i$ th cloud platform;  $CRSN_{ij}$  represents the  $j$ th candidate service resource that completes the  $i$ th sub-manufacturing task;  $d, s, k$  respectively represents the number of candidate resources in each candidate resource set.

#### 4. Matching Model

- (1) The cloud platform provides a set of candidate resources for manufacturing tasks requiring a single resource by analyzing the constraints of manufacturing resources and the basic attributes, state attributes, and functional attributes of the manufacturing resources. This process is a primary selection process of resource matching. The purpose is not to match a unique service resource for each manufacturing task that

- requires a single resource, but to search for a service resource whose condition meets the requirements, and to divide the resources that meet the user requirement into the candidate resource set from the massive service resource.
- (2) A manufacturing task requiring a single resource searches for a resource matching the user requirement in the candidate resource set, and determines the candidate service by analyzing the time  $T$ , quality  $Q$ , cost  $C$ , reliability  $Re$ , and recoverability  $G$  of the resource. Then, whether the resource matches the manufacturing task that requires a single resource is determined. If the resource can be matched, it is left; if there is no match, the candidate service resource is deleted from the candidate resource set. When the search match is over, the manufacturing resource that best meets the requirements to serve the user is selected.

#### 4.1. Matching Objective Function

The meanings of physical quantities are given in Table 1.

Table 1. Meaning of physical quantity.

Physical Quantity	Meaning
$CMR_i$	Candidate resource set
$CMTZ_i$	A single resource manufacturing task
$T_d$	Task acceptance time
$T_t$	Hard resources (transport time of physical resources such as device resources)
$T_w$	Soft resources (authorization of patent use, waiting time for debugging of software)
$T_s$	Resource service time
$\delta_w$	Waiting time factor
$\delta_t$	Transport time factor
$n$	Number of matching resources
$CRSN_{ij}$	The $j$ th candidate service resource for the $i$ th manufacturing task
$Q_i^j$	Evaluation of the $j$ th quality indicator after the user obtains the $i$ th resource
$\sum_{j=1}^m Q_h^j$	Quality of service for hard resources
$\sum_{j=1}^m Q_s^j$	Quality of service for soft resources
$\sum_{j=1}^m Q_o^j$	Quality of service for other resources
$\phi_h$	Quality of service factor for hard resources
$\phi_s$	Quality of service factor of soft resources
$\phi_o$	Quality of service factor for other resources
$M$	Number of user reviews
$C_h$	Cost of hard resources
$C_s$	Cost of soft resources
$C_h^k$	Fees paid after using hard resources
$C_h^t$	Shipping costs paid after using hard resources
$C_s^k$	Fees paid after using soft resources
$Re_b$	Resource failure rate
$Re_f$	Resource fault tolerance
$Re_s$	Resource security
$A$	the number of times that the service resource has resumed working after it stopped working due to an unexpected condition
$Z$	Number of unexpected conditions

**(1) Time  $T$** 

The service time is the time when the user issues a request command to the cloud platform, and the platform provides the corresponding service to the user according to the requirement. The service time  $T$  is mainly composed of four parts:  $T_d, T_t, T_w, T_s$ .

$$T = T_d + \delta_w T_w + \delta_t T_t + T_s \quad (1)$$

$$\delta_w = \begin{cases} 1 & \text{Waiting time} \\ 0 & \text{No waiting time} \end{cases}$$

$$\delta_t = \begin{cases} 1 & \text{Transportation time} \\ 0 & \text{No transportation time} \end{cases}$$

A manufacturing task that requires a single resource enters the candidate resource set and matches the resource in order. Therefore task-resource matching targets time function:

$$T = \sum_{j=1}^n T(CRSN_{ij}) \quad (2)$$

**(2) Quality  $Q$** 

Quality of service is the resource that the platform matches for users to meet the using requirements. The service quality is divided into  $\sum_{j=1}^m Q_h^j$  (assembly precision, dimensional accuracy, motion accuracy, processing scrap rate),  $\sum_{j=1}^m Q_s^j$  (operability, maintainability),  $\sum_{j=1}^m Q_o^j$  (customer service level, communication and cooperation level, logistics and transportation quality level).

$$Q = \sum_{j=1}^n (\phi_h Q_h^j + \phi_s Q_s^j + \phi_o Q_o^j) / M \quad (3)$$

$$\phi_h = \begin{cases} 1 & \text{Hard resources} \\ 0 & \text{No hard resources} \end{cases}$$

$$\phi_s = \begin{cases} 1 & \text{Soft resource quality} \\ 0 & \text{No soft resource quality} \end{cases}$$

$$\phi_o = \begin{cases} 1 & \text{Other resource quality} \\ 0 & \text{No other resource quality} \end{cases}$$

A manufacturing task that requires a single resource enters a candidate resource set to receive resources in order. Therefore task-resource matching targets quality function:

$$Q = \sum_{j=1}^n Q(CRSN_{ij}) \quad (4)$$

**(3) Cost  $C$** 

The platform provides users with corresponding resource services and users need to pay related fees. Service costs are divided into  $C_h$  and  $C_s$ .

For hard resource service costs:

$$C_h = C_h^k + C_h^t \quad (5)$$

For soft resource service costs:

$$C_s = C_s^k \quad (6)$$

A manufacturing task that requires a single resource enters a candidate resource set to receive resources in order, therefore task-resource matching targets cost function:

$$C = \sum_{j=1}^n C(CRSN_{ij}) \quad (7)$$

#### (4) Reliability $Re$

Reliability includes important evaluation parameters for reliable partnerships between users, platforms, and resource providers, to achieve long-term cooperation. Each resource entering the cloud platform will be accurately positioned by the cloud platform, and its corresponding reliability evaluation will be given, so that the user can obtain the required resources more accurately. Reliability evaluation mainly considers resource failure rate, resource fault tolerance, and resource security.

$$Re = (Re_b + Re_f + Re_s) / M \quad (8)$$

A manufacturing task that requires a single resource enters a candidate resource set to receive resources in order. Therefore, task-resource matching targets reliability function:

$$Re = \prod_{j=1}^n Re(CRSN_{ij}) \quad (9)$$

#### (5) Recoverability $G$

Resources that provide services to users can quickly resume normal working conditions after they stop working under certain special circumstances.

$$G = A / Z \quad (10)$$

A manufacturing task that requires a single resource enters a candidate resource set to receive resources in order. Therefore, task-resource matching targets reliability function:

$$G = \sum_{j=1}^n G(CRSN_{ij}) \quad (11)$$

### 4.2. Constraints

#### (1) Total time constraint

$$T_{\max} \geq \sum_{j=1}^n T(CRSN_{ij}) \quad (12)$$

When the manufacturing resource completes the task, the maximum time cannot exceed the maximum time period required by the user.

#### (2) Total quality constraint

$$Q_{\min} \leq \sum_{j=1}^n T(CRSN_{ij}) \quad (13)$$

When the manufacturing resources complete the corresponding task requirements, the quality of the manufactured product is not lower than the minimum value of the product quality requested by the user.

#### (3) Total cost constraint

$$C_{\max} \geq \sum_{j=1}^n T(CRSN_{ij}) \quad (14)$$

When the manufacturing resource completes the task, the cost of the requirement does not exceed the maximum cost required by the user.

#### (4) Reliability constraint

$$Re_{\min} \leq \prod_{j=1}^n Re(CRSN_{ij}) \quad (15)$$

The reliability of the service resources in the set of candidate resources participating in the manufacturing service is not lower than the minimum required by the user and the platform.

## (5) Recoverability constraint

$$G_{\min} \leq \sum_{j=1}^n G(CRSN_{ij}) \quad (16)$$

In the process of manufacturing resources in the service process, the probability of accidentally stopping work and returning to normal work is not lower than the probability that the user and the platform expect.

In the actual production process, users, platforms, and resource providers require different factors such as cost, time, and quality of products, and each factor is related to each other and affects each other. It is difficult to ensure that each objective function is optimal. Such multi-objective function optimization problems can be transformed into a single-objective optimization problem by linear weighted combination.

$$\min Y = w_1 \frac{T}{T_{\max}} + w_2 \frac{C}{C_{\max}} + w_3 \frac{Q_{\min}}{Q} + w_4 \frac{Re_{\min}}{Re} + w_5 \frac{G_{\min}}{G} \quad (17)$$

Among them,  $w_1$ ,  $w_2$ ,  $w_3$ ,  $w_4$ , and  $w_5$  are the weight coefficients of the five objective functions, respectively. The mathematical relationship is as follows:

$$w_1 + w_2 + w_3 + w_4 + w_5 = 1 \quad (18)$$

## 5. Case Analysis

With the rapid development of social productivity and increasingly fierce market competition, user demand is also characterized by diversity and complexity. However, due to various factors such as processing equipment, technology, talents, transportation, and information, it is difficult for enterprises to rely on their own resources to meet the diversity needs of users, so it is urgent to join other resources. The resource optimization matching simulation is carried out according to user requirements to verify the feasibility and practicability of the proposed method. Assuming that the user makes a task request on the cloud manufacturing platform, the process is now simulated. Through the task decomposition mechanism, the cloud platform decomposes the task request proposed by the user into  $CMTZ_1$ ,  $CMTZ_2$ ,  $CMTZ_3$ ,  $CMTZ_4$ ,  $CMTZ_5$ , and  $CMTZ_6$ . The six subtasks are filtered out from the massive resource pool according to the resource matching mechanism to form a set of candidate resources that meet the respective processing requirements, as shown in Table 2, and the data as shown in Table 3. According to the matching model of cloud manufacturing resources, the artificial bee colony algorithm is used to verify the correctness and efficiency of this model.

### 5.1. Determination of Weighting Factor by Analytic Hierarchy Process (AHP)

The model optimization problem in this paper is a multi-objective optimization problem. In order to search for the optimal solution more accurately, the multi-objective optimization problem is transformed into a single-objective optimization problem by using a linear weighted combination method. However, the target optimization problem is a mapping of real problems. All parameters have their physical meanings. Therefore, the selection of parameters will directly affect the final optimization result, so it is difficult for customers to determine a reasonable weighting factor as the coordination target coefficient. For the quantitative and qualitative analysis problem, the subjective factors account for a considerable proportion, and when the final optimization results are inconvenient, the analytic hierarchy process can effectively deal with such practical problems. The weighting factors of the main objective function in the model are calculated by the analytic hierarchy process, shown in Table 4.

**Table 2.** Candidate resource collections for subtasks.

Subtasks	CMTZ <sub>1</sub>	CMTZ <sub>2</sub>	CMTZ <sub>3</sub>	CMTZ <sub>4</sub>	CMTZ <sub>5</sub>	CMTZ <sub>6</sub>
Candidate resource set	CRSN <sub>1.1</sub>	CRSN <sub>2.11</sub>	CRSN <sub>3.26</sub>	CRSN <sub>4.32</sub>	CRSN <sub>5.41</sub>	CRSN <sub>6.51</sub>
	CRSN <sub>1.2</sub>	CRSN <sub>2.13</sub>	CRSN <sub>3.28</sub>	CRSN <sub>4.33</sub>	CRSN <sub>5.43</sub>	CRSN <sub>6.53</sub>
	CRSN <sub>1.4</sub>	CRSN <sub>2.17</sub>	CRSN <sub>3.29</sub>	CRSN <sub>4.34</sub>	CRSN <sub>5.44</sub>	CRSN <sub>6.55</sub>
	CRSN <sub>1.6</sub>	CRSN <sub>2.19</sub>	—	CRSN <sub>4.36</sub>	CRSN <sub>5.50</sub>	CRSN <sub>6.57</sub>
	CRSN <sub>1.7</sub>	—	—	CRSN <sub>4.39</sub>	—	CRSN <sub>6.58</sub>
	—	—	—	CRSN <sub>4.40</sub>	—	—
	—	—	—	—	—	—

**Table 3.** Relevant quantized data of candidate resources.

Resource	T	C	Q	Re	G
CRSN <sub>1.1</sub>	80	28	58.00	53.85	28
CRSN <sub>1.2</sub>	25	26	58.70	53.80	28
CRSN <sub>1.4</sub>	85	27	54.90	56.55	29
CRSN <sub>1.6</sub>	51	24	57.70	55.75	28
CRSN <sub>1.7</sub>	83	24	57.80	56.95	30
CRSN <sub>2.11</sub>	86	22	56.80	55.85	28
CRSN <sub>2.13</sub>	62	25	54.80	54.85	27
CRSN <sub>2.17</sub>	46	26	56.70	54.55	29
CRSN <sub>2.19</sub>	49	25	55.00	54.65	32
CRSN <sub>3.26</sub>	82	28	55.70	56.80	27
CRSN <sub>3.28</sub>	63	23	59.80	54.60	30
CRSN <sub>3.29</sub>	83	24	56.80	53.65	30
CRSN <sub>4.32</sub>	82	28	55.80	55.60	29
CRSN <sub>4.33</sub>	85	25	56.90	56.75	32
CRSN <sub>4.34</sub>	75	23	58.00	54.90	28
CRSN <sub>4.36</sub>	80	26	54.80	54.00	29
CRSN <sub>4.39</sub>	61	22	58.80	54.90	30
CRSN <sub>4.40</sub>	86	25	58.00	53.50	27
CRSN <sub>5.41</sub>	80	28	54.80	55.70	28
CRSN <sub>5.43</sub>	83	22	57.80	54.55	29
CRSN <sub>5.44</sub>	47	25	55.00	53.60	27
CRSN <sub>5.50</sub>	87	27	54.80	54.60	27
CRSN <sub>6.51</sub>	83	27	57.90	54.65	30
CRSN <sub>6.53</sub>	46	26	58.70	55.90	29
CRSN <sub>6.55</sub>	62	26	56.90	54.90	28
CRSN <sub>6.57</sub>	53	26	55.80	54.00	31
CRSN <sub>6.58</sub>	74	23	55.90	55.55	27

**Table 4.** Weighting factors.

Index	T	C	Q	Re	G
weighting	0.4237	0.2274	0.1422	0.1772	0.0294

### 5.2. Algorithm Solution

The artificial bee colony algorithm is a process that simulates a bee searching for food. The basic concepts include honey source and bee colony. The honey source is the manufacturing resource that meets the requirements of users and platforms. The bee colony includes hired bees, observation bees, and detection bees. Each hired bee corresponds to a certain nectar and searches for the field of nectar in an iteration. According to the size of the fitness value, the roe is used to hire the bee and the observation bee to search for the new honey source. If the honey source is updated many times without improvement, the honey source is discarded. The hired bees turn to detection to randomly search for new sources of honey. The artificial bee colony algorithm solves the matching model flow as shown in Figure 3.

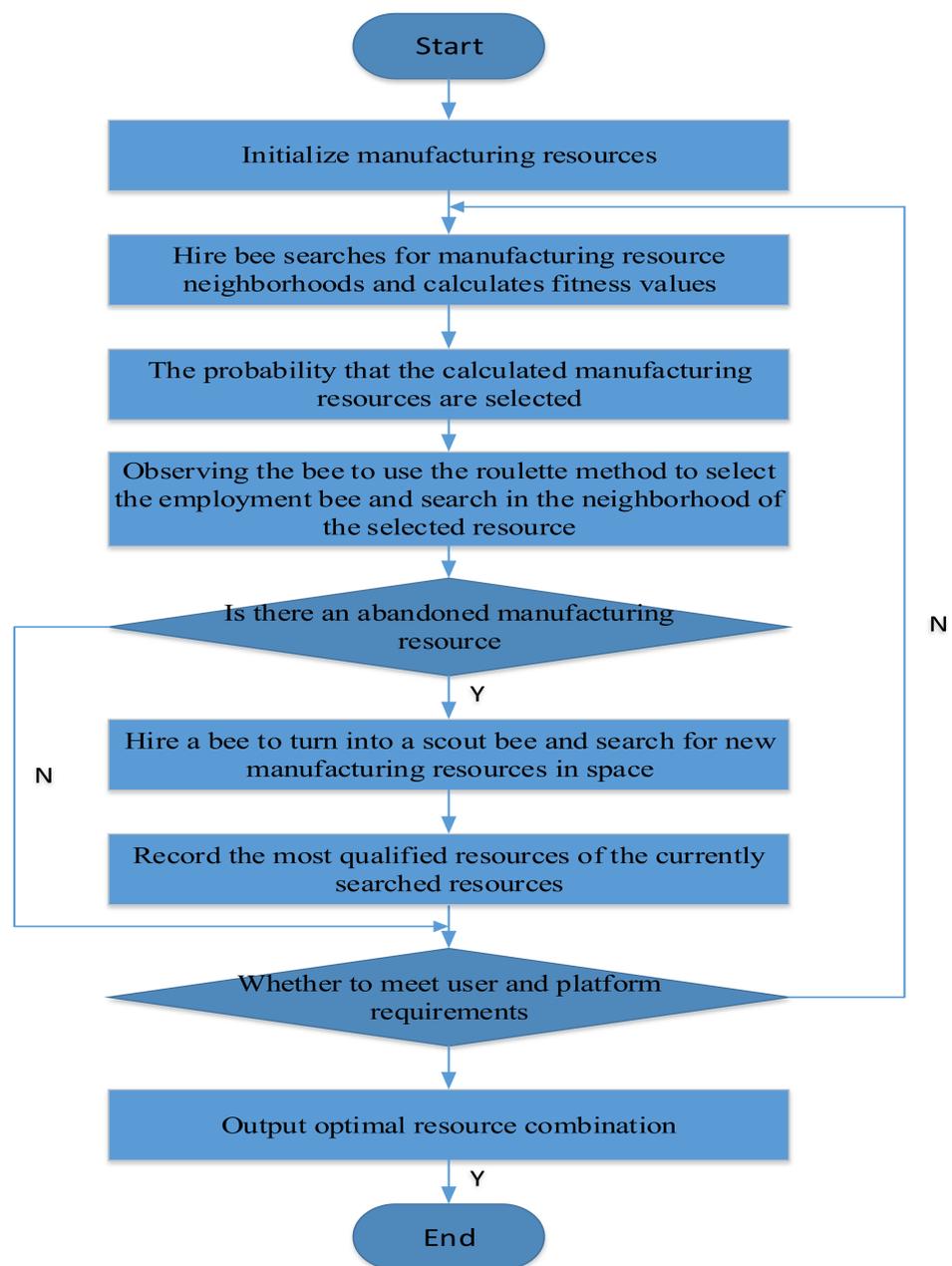


Figure 3. Artificial bee colony algorithm flow chart.

Based on the above artificial bee swarm intelligent algorithm solving model, the implementation is summarized as follows:

The solution to the mathematical model problem is five-dimensional. The number of hired bees and observed bees is set to 60, and the ABC algorithm considers the solution process of the optimization problem as searching in a five-dimensional space. According to the minimization problem of this paper, the algorithm solving process is realized: first, the manufacturing resource stage is randomly initialized, resource  $i$  meets  $\forall i \in \{1, 2, \dots, 60\}$ ,  $d \in \{1, 2, \dots, 5\}$ ,  $x_{id} \in (L_d, U_d)$ , where  $L_d$  represents the lower limit of the search space and  $U_d$  represents the upper limit of the search space. The position of the honey source  $i$  at the  $n$ th iteration is expressed as  $X_i^n = [x_{i1}^n, x_{i2}^n, \dots, x_{i5}^n]$ , the hired bee and the manufacturing resources are one-to-one correspondence, and the hired bee corresponding to the first resource searches for a new resource according to Equation (19).

$$x_{id} = L_d + rand(0, 1)(U_d - L_d) \quad (19)$$

Secondly, in the new honey source update search stage, the hired bee searches for a new honey source by searching for a neighborhood around the honey source  $i$  through Equation (20).

$$x'_{id} = x_{id} + \kappa(x_{id} - x_{ed}) \quad (20)$$

$i \in \{1, 2, \dots, 60\}$ ,  $d \in \{1, 2, \dots, 5\}$ ,  $e = \{1, 2, \dots, 60\}$  and  $e \neq i$  indicates that one of the 60 honey sources is randomly selected to be a honey source that is not equal to  $i$ .  $\kappa \in [-1, 1]$  represents a uniformly distributed random number, and determines the degree of disturbance. The algorithm brings the newly generated possible solution  $X'_i = \{x'_{i1}, x'_{i2}, \dots, x'_{i5}\}$  and the original solution  $X_i = \{x_{i1}, x_{i2}, \dots, x_{i5}\}$  into the fitness function  $f_i = w_1 \frac{T}{T_{\max}} + w_2 \frac{C}{C_{\max}} + w_3 \frac{Q_{\min}}{Q} + w_4 \frac{Re_{\min}}{Re} + w_5 \frac{G_{\min}}{G}$ . By comparing the fitness function values to preserve a better solution, all the hired bees complete the operation of Equation (20) and fly back to the hive to share the honey source.

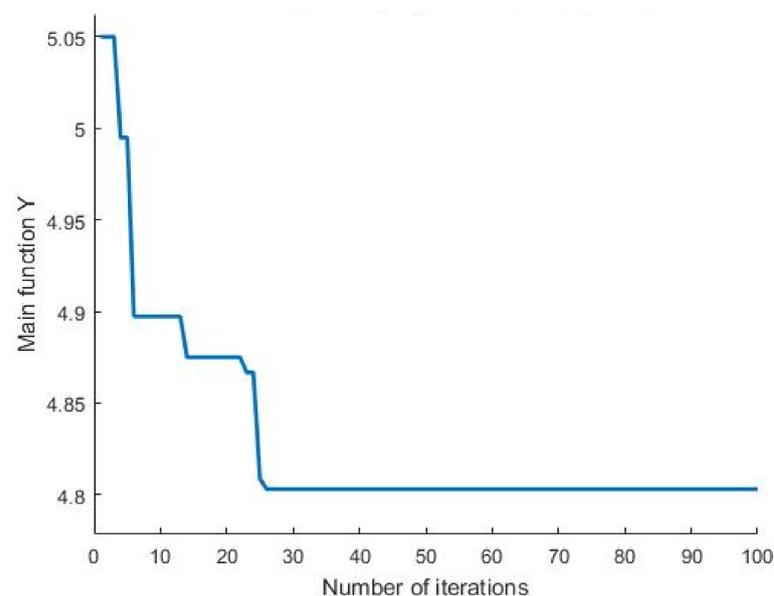
Then, the observer bee selects the hired bee, the observer bee updates the honey source by the roulette gambling method, and the probability can be calculated according to Equation (21), where,  $SN$  is the swarm size.

$$P_i = \frac{f_i}{\sum_{i=1}^{SN} f_i} \quad (21)$$

Finally, the detection bee stage is generated. After 100 iterations of the honey source, the optimal resource combination is found as shown in Table 5, and the objective function value and the iteration diagram are obtained as shown in Figure 4.

**Table 5.** Optimal resource combination.

Subtasks	CMTZ <sub>1</sub>	CMTZ <sub>2</sub>	CMTZ <sub>3</sub>	CMTZ <sub>4</sub>	CMTZ <sub>5</sub>	CMTZ <sub>6</sub>
Optimal resource	CRSN <sub>1.2</sub>	CRSN <sub>2.17</sub>	CRSN <sub>3.28</sub>	CRSN <sub>4.39</sub>	CRSN <sub>5.44</sub>	CRSN <sub>6.53</sub>



**Figure 4.** Iteration diagram.

It is known from the ABC algorithm iterative graph that the global optimal solution objective function  $Y = 4.803$  is obtained. In the 100 iterations, the ABC algorithm finds the convergence optimal solution after the 26th iteration.

## 6. Discussion

Cloud manufacturing is an intelligent, networked, and emerging manufacturing model with service-oriented and efficient utilization. Now, through the combination of

advanced network technology and cloud manufacturing technology, the manufacturing resources are virtualized and serviced, so that users and enterprises can effectively share and intelligently collaborate. Cloud manufacturing has a highly distributed layout and highly centralized resource usage, is service oriented and demand oriented, dynamic of manufacturing tasks, and involves user manufacturing and on-demand use and payment. Considering the above characteristics, it is found that the task and resource optimization matching in the cloud platform system is the most important and basic problem in the cloud platform. At present, the research results in this direction have different degrees of defects. For example, the resource matching model does not match the algorithm, and the resource matching and search efficiency are low. Based on the current research on resource matching direction, this paper proposes a new resource matching model, which is verified by a more appropriate heuristic algorithm. The detailed paper work is summarized as follows: First, based on the established cloud platform framework of task-resource matching, the dynamic search matching model with five indicators as parameters is designed in detail; the sub-tasks that require a single resource form a one-to-one mapping relationship with the candidate resource sets in the manufacturing resource pool, so that the resource-matching model can adapt to the complex heterogeneous resources in the cloud manufacturing. Second, through the optimization of the artificial bee colony algorithm, two types of tasks (subtasks requiring a single resource and total tasks requiring complex resources) can be well matched with service resources. Third, by establishing the relationship between different resources and tasks in each candidate resource set during the dynamic release of the task, and updating the information state of the resources in the cloud manufacturing resource pool, the matching result and efficiency of the task and the resource are not affected. This matching model is highly flexible and suitable for a variety of complex dynamic task matching problems in cloud manufacturing environments. Fourth, user engagement and user evaluation are fully considered in the task and resource matching model, which facilitates the updating and improvement of the manufacturing resource matching model, and fully reflects the fact that cloud manufacturing is user-oriented and service-oriented. Finally, the matching model extends to the later research. Through the redefinition of the data chain, matching index, and search rule of the intelligent matching model, it is helpful for scholars of different levels and different research directions to study.

## 7. Conclusions

Cloud manufacturing resource optimization matching is one of the most important issues in cloud manufacturing systems. Taking into account the complexity of the requirement task within of the resource matching process is important in order to provide users with more efficient and accurate service resources. Starting from the complexity of the task requirement resources, this paper transforms the tasks requiring multiple resources into a number of tasks requiring only a single resource, thus optimizing the matching of resources and reducing the difficulty of matching tasks and resources. Through the establishment of the task-resource matching model, and using the ABC algorithm to solve the model, the experiment proves the feasibility of the proposed model, and shows the high performance of the ABC algorithm in solving the resource optimization matching problem on the cloud platform.

The given methods are just a study and a simulation experiment created in this paper. Due to the complexity and dynamic nature of the demand tasks proposed by the user, in order to further improve the performance of the ABC algorithm, a more comprehensive simulation experiment design should be carried out for different demand tasks. How to find a feasible solution that meets all the needs of users in a short time will become an important issue. Therefore, a future research trend can be to optimize the matching of resources from research and design more efficient algorithms.

**Author Contributions:** Conceptualization, Y.H.; methodology, Z.W.; software, L.P.; validation, H.L.; writing—original draft preparation, Y.W.; writing—review and editing, Y.H.; visualization, D.G.;

supervision, Y.H.; project administration, Y.H.; funding acquisition, Y.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

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

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** This research work was supported by the Education Department Project of Jilin Province, no. JJKH20200659KJ.

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

## References

- Li, X.; Zhuang, P.; Yin, C. A metadata based manufacturing resource ontology modeling in cloud manufacturing systems. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 1039–1047. [\[CrossRef\]](#)
- Fu, J. A practical resource-searching method for manufacturing grid. *Int. J. Adv. Manuf. Technol.* **2014**, *74*, 335–340. [\[CrossRef\]](#)
- Li, B.H.; Zhang, L.; Wang, S.L.; Tao, F.; Cao, J.W.; Jiang, X.D.; Song, X.; Chai, X.D. Cloud manufacturing: A new service-oriented networked manufacturing model. *Comput. Integr. Manuf. Syst.* **2010**, *16*, 1–7. (In Chinese) [\[CrossRef\]](#)
- Zhang, L.; Luo, Y.L.; Tao, F.; Ren, L.; Guo, H. Key technologies for the construction of manufacturing cloud. *Comput. Integr. Manuf. Syst.* **2010**, *16*, 2510–2520. (In Chinese) [\[CrossRef\]](#)
- Zhou, J.; Wang, M. Cloud Manufacturing Service Paradigm for Group Manufacturing Companies. *Adv. Mech. Eng.* **2014**, *6*, 740725. [\[CrossRef\]](#)
- Jin, H.; Fu, Y.; Yang, G.; Zhu, X. An intelligent scheduling algorithm for resource management of cloud platform. *Multimed. Tools Appl.* **2020**, *79*, 5335–5353. [\[CrossRef\]](#)
- Gong, X.; Yin, C.; Li, X. A grey correlation based supply–demand matching of machine tools with multiple quality factors in cloud manufacturing environment. *Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 1025–1038. [\[CrossRef\]](#)
- Cheng, Y.; Tao, F.; Zhao, D.M.; Zhang, L. Modeling of manufacturing service supply–demand matching hyper network in service-oriented manufacturing systems. *Robot. Comput.-Integr. Manuf.* **2017**, *45*, 59–72. [\[CrossRef\]](#)
- Pan, X.Y.; Ma, J.Z.; Zhao, D.Z. Study on pricing behaviour and capacity allocation of cloud manufacturing service platform. *Clust. Comput.* **2019**, *22*, 14701–14707. [\[CrossRef\]](#)
- Laili, Y.; Tao, F.; Zhang, L.; Sarker, B.R. A study of optimal allocation of computing resources in cloud manufacturing systems. *Int. J. Adv. Manuf. Technol.* **2012**, *63*, 671–690. [\[CrossRef\]](#)
- Sheng, B.; Zhang, C.; Yin, X.; Lu, Q.; Cheng, Y.; Xiao, T.; Liu, H. Common intelligent semantic matching engines of cloud manufacturing service based on OWL-S. *Int. J. Adv. Manuf. Technol.* **2016**, *84*, 103–118. [\[CrossRef\]](#)
- Morariu, O.; Borangiu, T.; Morariu, C.; Răileanu, S. Service Oriented Mechanisms for Smart Resource Allocation in Private Manufacturing Clouds. In Proceedings of the International Conference on Exploring Services Science, Bucharest, Romania, 25–27 May 2016. [\[CrossRef\]](#)
- Zheng, H.; Feng, Y.; Tan, J. A fuzzy QoS-aware resource service selection considering design preference in cloud manufacturing system. *Int. J. Adv. Manuf. Technol.* **2016**, *84*, 371–379. [\[CrossRef\]](#)
- Guo, Y.; Xie, S. Modeling Method of Resource Combination Optimization for Crowdsourcing Product Development Based on Cloud. In Proceedings of the International Conference on Mechanical Design, Beijing, China, 13–15 October 2017. [\[CrossRef\]](#)
- Hu, Y.J.; Chang, X.F.; Wang, Y.; Wang, Z.L.; Shi, C.; Wu, L.Z. Cloud manufacturing resources fuzzy classification based on genetic simulated annealing algorithm. *Mater. Manuf. Process.* **2017**, *32*, 1109–1115. [\[CrossRef\]](#)
- Tao, F.; Li, C.; Liao, T.W.; Laili, Y. BGM-BLA: A New Algorithm for Dynamic Migration of Virtual Machines in Cloud Computing. *IEEE Trans. Serv. Comput.* **2016**, *9*, 910–925. [\[CrossRef\]](#)
- Cao, Y.; Wu, Z.; Liu, T.; Gao, Z.; Yang, J. Multivariate process capability evaluation of cloud manufacturing resource based on intuitionistic fuzzy set. *Int. J. Adv. Manuf. Technol.* **2016**, *84*, 227–237. [\[CrossRef\]](#)
- Guo, L.; Qiu, J. Optimization technology in cloud manufacturing. *Int. J. Adv. Manuf. Technol.* **2018**, *97*, 1181–1193. [\[CrossRef\]](#)
- Li, F.; Zhang, L.; Liu, Y.K.; Laili, Y.; Tao, F. A clustering network-based approach to service composition in cloud manufacturing. *Int. J. Comput. Integr. Manuf.* **2017**, *30*, 1331–1342. [\[CrossRef\]](#)
- Song, T.; Liu, H.; Wei, C.; Zhang, C. Common engines of cloud manufacturing service platform for SMEs. *Int. J. Adv. Manuf. Technol.* **2014**, *73*, 557–569. [\[CrossRef\]](#)
- Zhou, L.F.; Zhang, L.; Laili, Y.J.; Zhao, C.; Xiao, Y.Y. Multi-task scheduling of distributed 3D printing services in cloud manufacturing. *Int. J. Adv. Manuf. Technol.* **2018**, *96*, 3003–3017. [\[CrossRef\]](#)
- Tasgetiren, M.F.; Pan, Q.K. A discrete artificial bee colony algorithm for the total flowtime minimization in permutation flow shops. *Inf. Sci.* **2011**, *181*, 3459–3475. [\[CrossRef\]](#)
- Li, H.; Li, W. Enhanced artificial bee Colony algorithm and its application in multi-threshold image feature retrieval. *Multimed. Tools Appl.* **2019**, *78*, 8683–8698. [\[CrossRef\]](#)