sustainability

MDPI

Article

Multi-Objective Service Selection and Scheduling with Linguistic Preference in Cloud Manufacturing

Wei He [1], Guozhu Jia [1,*], Hengshan Zong [2] and Jili Kong [3]

[1] School of Economics and Management, Beihang University, Beijing 100191, China; heweibuaa@buaa.edu.cn
[2] Institute of Systems Engineering, China Aerospace Academy of Systems Science and Engineering, Beijing 100048, China; zy1708114@buaa.edu.cn
[3] School of Modern Post, Beijing University of Posts and Telecommunications, Beijing 100876, China; kongjili1026@bupt.edu.cn
* Correspondence: jiaguozhu@buaa.edu.cn; Tel.: +86-10-8231-7834

Received: 1 April 2019; Accepted: 30 April 2019; Published: 7 May 2019

check for updates

Abstract: Service management in cloud manufacturing (CMfg), especially the service selection and scheduling (SSS) problem has aroused general attention due to its broad industrial application prospects. Due to the diversity of CMfg services, SSS usually need to take into account multiple objectives in terms of time, cost, quality, and environment. As one of the keys to solving multi-objective problems, the preference information of decision maker (DM) is less considered in current research. In this paper, linguistic preference is considered, and a novel two-phase model based on a desirable satisfying degree is proposed for solving the multi-objective SSS problem with linguistic preference. In the first phase, the maximum comprehensive satisfying degree is calculated. In the second phase, the satisfying solution is obtained by repeatedly solving the model and interaction with DM. Compared with the traditional model, the two-phase is more effective, which is verified in the calculation experiment. The proposed method could offer useful insights which help DM balance multiple objectives with linguistic preference and promote sustainable development of CMfg.

Keywords: cloud manufacturing; service selection and scheduling; linguistic preference; multi-objective optimization; genetic algorithm

1. Introduction

With extensive application of the Internet, big data, and cloud computing in industry, cloud manufacturing (CMfg), a new service-oriented business model, was proposed in 2010 [1,2]. In the last ten years, it has received more and more attention not only from industrial enterprises but also from researchers [3–6]. With the support of information technology, CMfg is designed to realize sharing of resources distributed in different places [7–11]. In order to realize its objective, distributed manufacturing resources are aggregated by a common cloud platform and encapsulated into different kinds of manufacturing services [12,13]. These virtual services will be provided to clients or users in the pay-as-you-go mode. Given the various needs of clients, flexible selection and scheduling of these services become a significant challenge.

Generally, CMfg platform is operated in the following two manners [14,15]. The first one is decentralized operating mode, in which clients can directly select the available services on the platform and pay the services in need. In such mode, the service selection and scheduling (SSS) decisions are made by clients independently. Another one is centralized operating mode, in which cloud platform selects and schedules the services to satisfy clients' requests, and clients only need to present their demands and expectations. In the current customization and individualization development trend of industrial product, manufacturing tasks have become more and more complex. The centralized operation mode has higher control over distributed resources and is more conducive to handling complex tasks.

So far, a number of researchers have studied service selection problem or service/task scheduling problem in CMfg with a centralized operation mode. Akbaripour et al. [16] proposed different models for the basic service composition structures (i.e., sequential, parallel, loop, selective). In their models, QoS metric consists of cost, time, and quality. Cheng et al. [17] consider comprehensive utility composed of energy consumption, cost, and risk. Then four kinds of resource service scheduling modes were studied. Liu et al. [15] present a scheduling model considering the workload of each task. In general, the attributes or criteria, which have been considered in previous works, could be divided into: cost-related indices [5,16,18,19], time-related indices [5,15,16,18,19], quality-related indices [5,16], risk-related indices [5], reliability-related indices [5,18,19], trust related indices [5,20], environment-related indices [21]. In addition, some others focus on the factors that can reflect the characteristics of cloud manufacturing, such as demand loss probability [22], correlation [23], and tolerance design [24].

In the above studies, all of them are based on single-objective optimization [15] or convert multi-objective problems into single-objective optimization [5,18,25]. Multi-objective optimization is also a concern. Xiang et al. [19] introduced group leader algorithm (GLA) into the service composition problem, and proposed a GLA-Pareto method. Even though Pareto solution set is found, decision makers (DM) still need to choose the optimal solution based on their preference. The articulation of preference information can be divided into three categories: priori, progressive, and posteriori. The priori articulation is that the DM provides the preference information before solving the problem [26]. As is typical in a priori articulation, weighted parameters are widely used to control the relative importance of each objective [11,18]. Resources on the CMfg platform can dynamically join and exit, which increases the difficulty for DM to specify the weight of each objective. In addition, the optimization satisfying degree of the objective does not necessarily coincide with the importance exhibited by the accurate weight value. As far as we know, in previous studies, the weight values are usually given directly, ignoring the specific definition process of weight. This issue correspondingly motivates us to consider other kinds of preference information of DM for the SSS problem in CMfg. To assess the preference of DM, Narasimhan [27] has introduced linguistic terms. Chen and Tsai [28] put forward the principle of transforming the importance of objectives into inequality constraints. Based on their idea, this paper will investigate how to choose the most preferred solution for multi-objective SSS problem if the relative importance is assessed by linguistic terms.

To date, there has been relatively little attention devoted to preference information of DM in CMfg. This paper intends to consider linguistic preference in the SSS problem. In particular, the purpose of this paper is to explain three questions: (1) How to express the linguistic preferences of decision makers in the SSS model? (2) How to maximize the difference in satisfying degree while achieving the overall optimization of all objectives? (3) How does DM participate in the SSS process and choose a satisfying solution?

The rest of this paper is organized as follows. Section 2 gives a brief description of the problem. Section 3 elaborates on the proposed SSS model and solution methods in detail. Section 4 shows and analyzes the results of computational experiments. Finally, in Section 5, conclusions and future research directions are described.

2. Problem Description

This paper considers a CMfg platform with centralized operating mode, and the SSS is system-centered. The framework of centralized operating mode for CMfg platform is shown in Figure 1 [16]. Suppose that there are M enterprises, expressed as $E = \{E_1, \ldots, E_i, \ldots, E_M\}$. Each enterprise provides several services. For example, $SE_i = \left\{SE_{i,1}, \ldots, SE_{i,h}, \ldots, SE_{i,H_i}\right\}$ is the service set of the enterprise E_i, where $SE_{i,h}$ represent the hth service, and H_i is the total number of service provided by E_i.

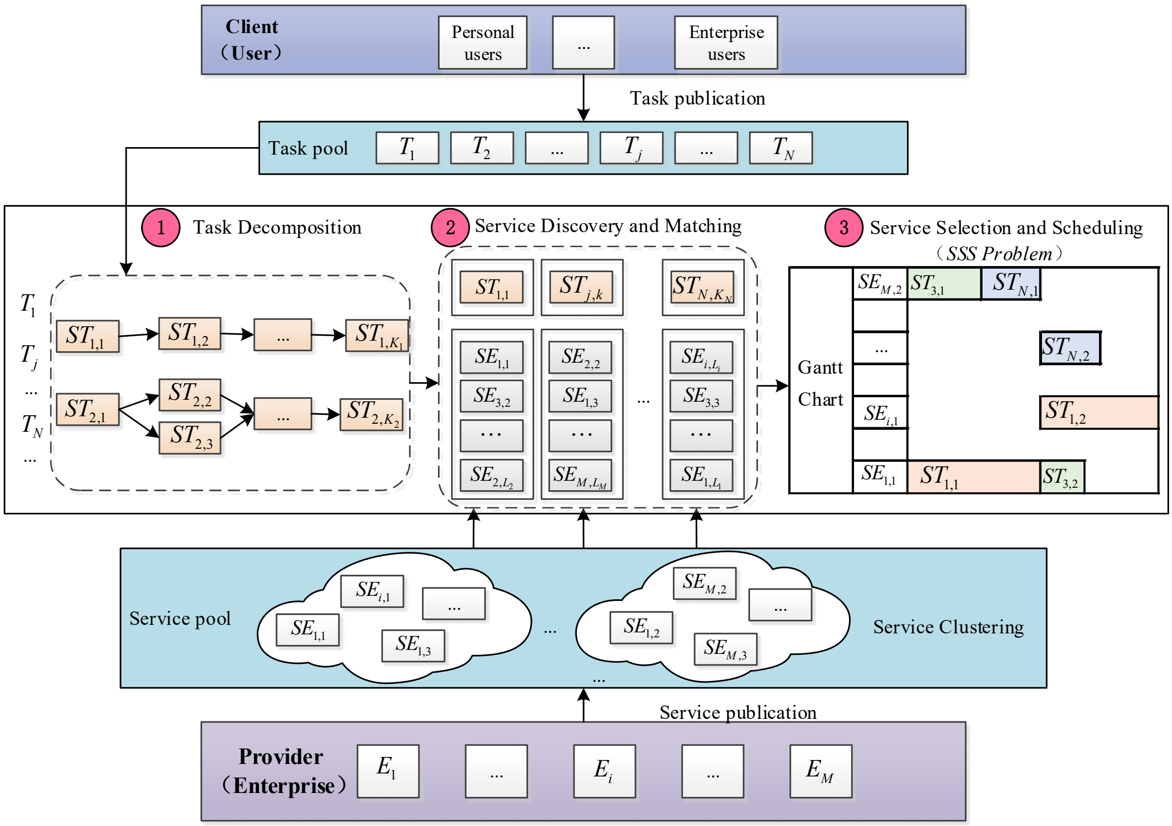

Figure 1. The framework of centralized operating mode for cloud manufacturing (CMfg) platform.

Clients can submit orders to the CMfg platform, and each order that meets the basic requirements of the platform will be accepted and considered as a task. In order to complete these accepted tasks, the platform mainly performs the following three processes [16]:

- Task decomposition: Manufacturing tasks in CMfg can be divided into simple tasks and complex tasks. Here, simple tasks can be assigned directly to available services, while complex tasks have to be decomposed into multiple subtasks, so that each subtask can be performed by a separate service [5,16]. There are four commonly accepted composition structures: sequential, parallel, selective, and loop [16,18,23].
- Service discovery and matching: for each subtask, the available services are found and put into the service set.
- Service selection and scheduling (SSS): for each subtask, one service is chosen from the corresponding service set, then these subtasks are scheduled onto the available time windows of selected services, and routes required transportations, such that the overall objectives are optimized.

Suppose there are N tasks received on a platform within a certain decision period to be processed, which are represented by $T = \{T_1, \ldots, T_j, \ldots, T_N\}$. Each task contains multiple subtasks, and for task T_j, the subtask set is represented as $ST_j = \{ST_{j,1}, ST_{j,2}, \ldots, ST_{j,K_j}\}$.

The objective function of SSS problem usually requires consideration of criteria, such as cost, time, quality, and environment. For the system-centered SSS, the choice of the final solution depends on the preference information given by decision maker (DM). Linguistic terms, as one of the power descriptions for preference, are considered in this paper. For the convenience of modeling, the linguistic terms are expressed by the importance parameter b. Here, we consider four linguistic terms "general" ($b = 4$), "important" ($b = 3$), "somewhat important" ($b = 2$), and "very important" ($b = 1$). Then, the key problem becomes how to choose the most preferred solution, which optimizes multiple objectives in collaboration, while reflecting the relative importance based on linguistic preference.

3. The Multi-Objective Service Selection and Scheduling Model and Solution Methods

3.1. Assumptions

- All tasks are independent of each other.
- The service capabilities have been fully gathered in the given period.
- Each subtask has qualified service set and must be assigned to one service to complete.
- A started subtask cannot be interrupted.
- Before the SSS process, the service pool for each subtask has been built. Moreover, the time, cost, quality, and environmental cost of service for different services are already known.
- Only the sequence model is considered in this paper, which is just to simplify the calculation and does not affect the results of the study. Then, there is a precedence constraint relationship between various subtasks in a task.
- Consider a situation where there is only one DM.

3.2. Notations

$st_{j,k}^{i,h}$ Service time of subtask $ST_{j,k}$, if subtask $ST_{j,k}$ is assigned to service $SE_{i,h}$.
$sc_{j,k}^{i,h}$ Service cost of subtask $ST_{j,k}$, if subtask $ST_{j,k}$ is assigned to service $SE_{i,h}$.
$q_{j,k}^{i,h}$ Service quality of subtask $ST_{j,k}$, if subtask $ST_{j,k}$ is assigned to service $SE_{i,h}$.
$ec_{j,k}^{i,h}$ Environmental cost of subtask $ST_{j,k}$, if subtask $ST_{j,k}$ is assigned to service $SE_{i,h}$.
$we_{j,k}^{i,h}$ Weight of products needed to be transported, if subtask $ST_{j,k}$ is assigned to service $SE_{i,h}$.
$at_{j,k}$ Start time of subtask $ST_{j,k}$.
$ct_{j,k}$ Completion time of subtask $ST_{j,k}$.
$wt_{j,k}$ Waiting time of subtask $ST_{j,k}$.
$lt_{j}^{k,k+1}$ Logistics time from subtask $ST_{j,k}$ to $ST_{j,k+1}$.
$lc_{j}^{k,k+1}$ Logistics cost from subtask $ST_{j,k}$ to $ST_{j,k+1}$.
$d_{i,i'}$ Geographical distance between enterprises E_1 and E_2.
α Logistics time for unit distance.
β Logistics cost for unit distance and unit weight.

3.3. Model Formulation

For the multi-objective optimization problem, a general model is expressed as:

$$\begin{cases} \min f_1, f_2, \ldots, f_g \\ \max f_{g+1}, f_{g+2}, \ldots, f_G \\ \quad s.t. \quad x \in X_d, X_d = \{x | y_i(x) \le b_i, i = 1, 2, \ldots, m\} \end{cases} \tag{1}$$

In which $f_1, f_2, \ldots, f_g$ are the negative objectives for minimization, such as cost, time, etc. and $f_{g+1}, f_{g+2}, \ldots, f_G$ are the positive criteria or objectives for maximization like quality, and reliability. X_d is the feasible solution set.

As an optimization problem, SSS needs to consider the construction of an objective function. In previous works, various types of criteria for service selection or service/task scheduling have been presented. Considering the sustainability of the CMfg, the total environmental cost and some other system-centered criteria, such as maximum completion time, total service cost, average quality for all tasks, are considered in this paper. Then the multi-objective model in cloud manufacturing is as follows:

$$\min f_1 = \max\left(t_j\right) = \max\left(st_j + lt_j + wt_j\right)(j = 1, 2, \ldots, N) \tag{2}$$

$$\min f_2 = \sum_{j=1}^{N} c_j = \sum_{j=1}^{N}\left(sc_j + lc_j\right) \quad (3)$$

$$\max f_3 = \left(\sum_{j=1}^{N} q_j\right)/N \quad (4)$$

$$\min f_4 = \sum_{j=1}^{N} ec_j \quad (5)$$

Subject to:

$$t_j \le t_j^{\max}, \forall j = 1, \ldots, N; \quad (6)$$

$$c_j \le c_j^{\max}, \forall j = 1, \ldots, N; \quad (7)$$

$$q_j \ge q_j^{\min}, \forall j = 1, \ldots, N; \quad (8)$$

$$ec_j \le ec_j^{\max}, \forall j = 1, \ldots, N; \quad (9)$$

$$\sum_{i=1}^{M} \sum_{h=1}^{H_i} x_{j,k}^{i,h} = 1, \forall j = 1, \ldots, N; k = 1, \ldots, K_j; \quad (10)$$

$$x_{j,k}^{i,h} \in \{0,1\}, \forall j = 1, 2, \ldots, N; i = 1, \ldots, M; k = 1, \ldots, K_j; h = 1, \ldots, H_i; \quad (11)$$

where, $x_{j,k}^{i,h} = 1$ if subtask $ST_{j,k}$ is assigned to service $SE_{i,h}$, otherwise $x_{j,k}^{i,h} = 0$. If $x_{j,k}^{i,h} = 1$, then completion time of subtask $ct_{j,k} = at_{j,k} + st_{j,k}^{i,h}$, Logistics time $lt_j^{k,k+1} = \alpha \times d_{ii'}$, Logistics cost $lc_j^{k,k+1} = \beta \times we_{j,k}^{i,h} \times d_{ii'}$. Objectives (2)–(5) are respectively the maximum completion time f_1, total service cost f_2, average quality f_3, and total environmental cost f_4 for all tasks. For each task T_j, its completion time t_j, task cost c_j, and service quality q_j should not exceed the limits given by the client, which is represent by $t_j^{\max}$, $c_j^{\max}$,$q_j^{\min}$ in Constraints (6)–(8). In our model, we assume that subtask $ST_{j,k}$ can be assigned to only one service $SE_{i,h}$. This can be described by Formulas (10) and (11).

In a real situation, multiple objectives might conflict with each other and cannot be optimized simultaneously. Then, the membership function μ_{f_g} for the gth objective is defined and expressed as:

$$\mu_{f_g} = \begin{cases} 1 & f_g \le f_g^{\min} \\ 1 - \left(f_g - f_g^{\min}\right)/\left(f_g^{\max} - f_g^{\min}\right) & f_g^{\min} < f_g \le f_g^{\max} \\ 0 & f_g > f_g^{\max} \end{cases} \quad (12)$$

$$\mu_{f_g} = \begin{cases} 1 & f_g \ge f_g^{\max} \\ \left(f_g - f_g^{\min}\right)/\left(f_g^{\max} - f_g^{\min}\right) & f_g^{\min} \le f_g < f_g^{\max} \\ 0 & f_g < f_g^{\min} \end{cases} \quad (13)$$

where, Formula (12) is for minimization objectives and Formula (13) is for maximization objectives. $f_g^{\min}$ and $f_g^{\max}$ can be obtained by solving each objective or given by DM. Normalization reduces the impact of a single objective on other objectives in the optimization process. Then the objective function is expressed as:

$$\max\left(\mu_{f_1}, \mu_{f_2}, \mu_{f_3}, \mu_{f_4}\right) \quad (14)$$

Faced with the situation that decision maker can only give linguistic preference information rather than exact weights. According to Chen and Tsai's [28] ideas, the relative importance could be expressed as:

$$\mu_{f_g} \ge \mu_b^*, \forall P\left(f_g\right) = b \quad (15)$$

where $P\left(f_g\right) = b$ mean that the gth objective's relative importance is b, the corresponding desirable satisfying degree is μ_b^*.

If the Formula (15) is directly given by the DM and added to the constraints of SSS problem, the feasible region will be greatly reduced, especially when the desirable satisfying degree is too high.

Therefore, we have to find a better solution on the basis of (15). Instead of requiring the DM to give μ_b^* beforehand, this paper treats it as a variable. The principle that more important objectives have a higher desirable satisfying degree is expressed as:

$$\mu_b^* \geq \mu_{b'}^*, \forall b < b' \tag{16}$$

However, Formula (16) only shows the different relative importance, and its strict comparative relation is still not conducive to finding a more satisfactory solution. Therefore, in order to avoid this situation, the variable γ is introduced to relax it, forming the following importance comparison relationship:

$$\mu_b^* - \mu_{b'}^* \geq \gamma, \forall b < b' \tag{17}$$

If $\gamma \geq 0$, the importance requirement is met, else if $\gamma < 0$, it does not meet the importance requirement.

In order to balance optimization of all objectives with different importance, and to avoid strict Constraint (16) which may lead to no solution, we decompose the SSS problem with linguistic preference into two sub-problems. The satisfying degree of the optimization results requires the participation of the DM. Therefore, the two-phase interactive optimization model is constructed as follows:

Phase 1:

$$\begin{cases} \max & \lambda \\ s.t. & \mu_{f_g} \geq \lambda, g = 1,2,3,4 \\ & \mu_{f_g} \leq 1 \\ & (2) \sim (5) \\ & (11) and (12) \\ & (6) \sim (10) \end{cases} \tag{18}$$

where $x_{j,k}^{i,h}$ are the decision variables. The optimal solution λ^* is defined as the maximum comprehensive satisfying degree, which represents the maximum satisfying degree value that can be achieved by the worst objective of all objectives without considering preference. It should be noted that the constraint $\mu_{f_i}(x) \leq 1$ ensures comparability between satisfying degree of different objectives.

Phase 2:

Desirable satisfying degree is divided into different levels by Formula (17). In order to expand the scope of feasible region, we relax the maximum comprehensive satisfying degree λ^*.

$$\mu_{f_g} \geq \mu_b^* \geq \lambda^* * \Delta\delta, \forall P\left(f_g\right) = b \tag{19}$$

where μ_b^* is taken as a variable, $\Delta\delta (0 \leq \Delta\delta \leq 1)$ is the parameter to relax the maximum comprehensive satisfying degree λ^*. Based on the Constraint (19), the second model is established as:

$$\begin{cases} \max & \gamma \\ s.t. & \mu_{f_g} \geq \mu_b^* \geq \lambda^* * \Delta\delta, \forall P\left(f_g\right) = b, g = 1,2,3,4, b = 1,2,3,4 \\ & \mu_b^* - \mu_{b'}^* \geq \gamma, \forall b < b', b, b' = 1,2,3,4 \\ & (2) \sim (5) \\ & (11) and (12) \\ & \mu_{f_i}(x) \leq 1 \\ & 0 \leq \gamma \leq 1 \\ & (6) \sim (10) \end{cases} \tag{20}$$

where $x_{j,k}^{i,h}$, μ_b^*, γ are the decision variables. As an optimization indicator, maximizing γ means to maximize differences in desirable satisfying degrees among objectives of different importance.

3.4. Optimization Algorithm

Step 1: Set up the membership functions of the objectives based on the requirement of DM.

Step 2: Calculate the maximum comprehensive satisfying degree λ^* based on Formula (18)

Step 3: Set $\Delta\delta = 0$, list the same comparative relationships as Formula (19) according to the linguistic preferences of DM, then construct the second Model (20).

Step 4: Solve Model (20) by a suitable single objective method.

Step 5: If no feasible solution is found, then go to step 6. On the contrary, if a feasible solution is found, the DM will decide whether the solution is satisfying or not. If $\gamma < 0$ or $\gamma > 0$ but DM is not satisfied, then go to step 6. If it does not belong to the above situations, then a satisfactory solution has been found.

Step 6: Set the new parameter $\Delta\delta$ to relax λ^*, and go back to step 4.

4. Computational Experiments and Results

A small example is first used to demonstrate the working of the two-phase method. Then a number of computational experiments with problems of various sizes are designed to demonstrate its effectiveness and efficiency. At last, the performance stability of the proposed method is tested.

4.1. A Small Scheduling Example

Figure 2 presents a small SSS example in system-centered CMfg with centralized operating mode. In this example, three tasks are considered, and each consists of four subtasks. Assume that there are two enterprises on the CMfg platform, i.e., E_1 to E_2. Every enterprise provides two types of service. The detail of the example is shown in Appendix A. The task and service information is shown in Table A1, and the geographical distance between enterprises is shown in Table A2. To test the validity of the two-phase method, the max-min method and the weighted sum method are adopted for comparison. DM has different preferences for multiple objectives. For the two-phase method, the relative importance of the objectives is 3-2-4-1. $b_4 = 1$ means that the fourth objective is the most important, $b_3 = 4$ means that the third objective is the most unimportant. For the weighted sum method, we invited 10 volunteers to be decision makers. They gave the weight of each objective according to the relative importance (3-2-4-1). Then the average values of these 10 groups of weights were calculated and used as final weights. The final objective weights are 0.143, 0.286, 0.095, 0.476, respectively. It should be pointed out that the max-min method and the weighted sum method do not need to find the desirable satisfying degree. However, in order to make a comparative analysis, the optimal solutions obtained by the two methods are added to Model (20) as a known condition to find the corresponding μ_b^* and γ. $\Delta\delta$ in the three method are all set as 0.9.

Three best feasible SSS results found by the above mentioned three SSS methods are shown in Figure 2. We can see that the completion time t_j of task T_j is made up of three parts: logistics time lt_j, service time st_j and waiting time wt_j. Through the start time at_{jk} and the service time st_{jk}^{ih}, the completion time of subtask ST_{jk} could be found by the formula $ct_{jk} = at_{jk} + st_{jk}^{ih}$. The figure also shows three methods to find different completion times.

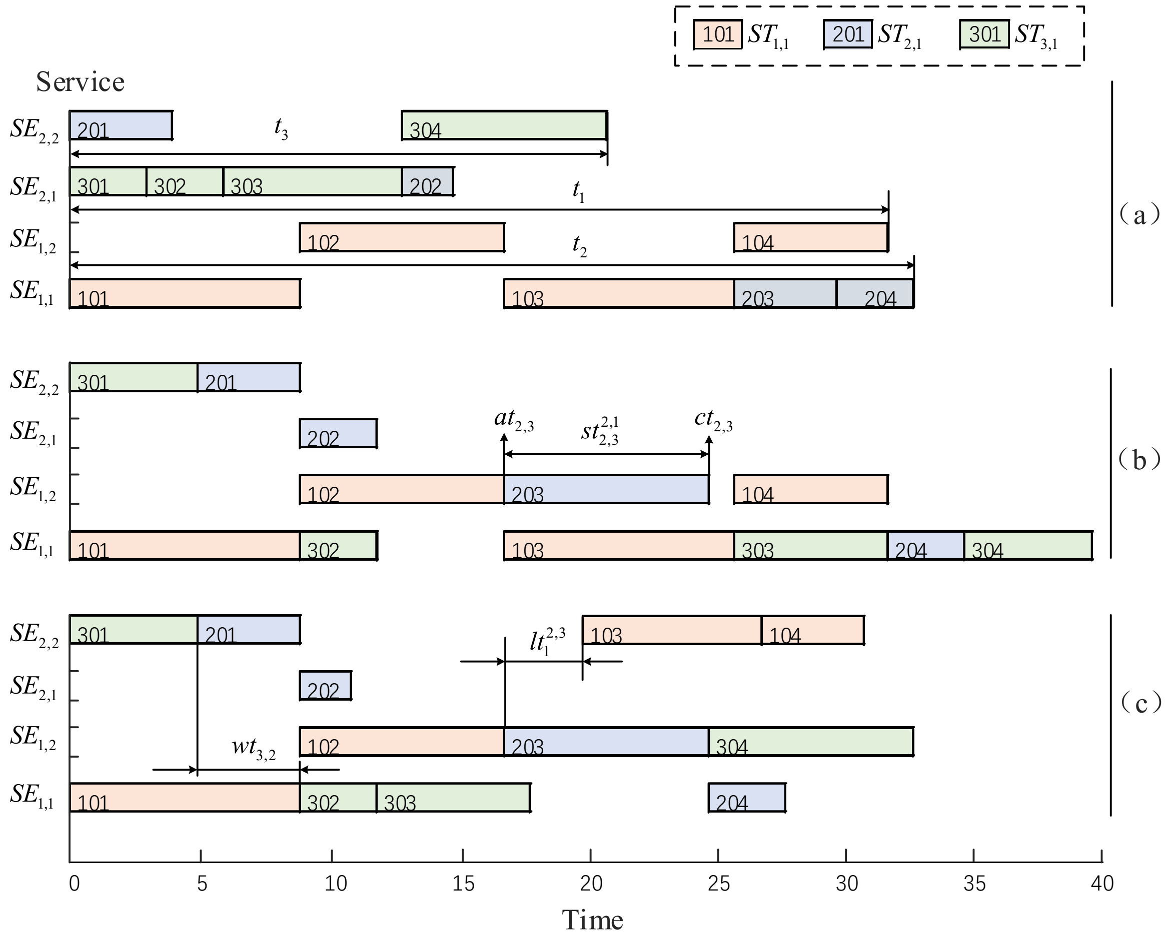

Figure 2. Best feasible service selection and scheduling (SSS) results of the problem in Section 4.1 found by (**a**) max-min method (**b**) two-phase method (**c**) weighted sum method.

Figure 3 shows the satisfying degree and desirable satisfying degree corresponding to the best feasible SSS results. From the perspective of actual satisfying degree μ_{f_g}, the difference of satisfying degree in (a) is not consistent with the relative importance of the objectives (3-2-4-1), and the minimum satisfying degree $\mu_{f_g}^{\min} = \min\left(\mu_{f_g}\right) = 0.5714$ is the largest of the three methods. The difference of satisfying degree in (c) is the most obvious, however it does not fully conform to "objectives with higher importance have greater satisfying degree values." Compared with (a) and (c), the satisfying degree in (b) not only achieves the overall optimization, but also maintains the important difference. The concept of redundant satisfying degree (*rs*) is introduced to compare the three methods from the perspective of a desirable satisfying degree μ_b^*, which means the difference between an actual satisfying degree and a corresponding desirable satisfying degree. For the gth objective, $rs_g = \mu_{f_g} - \mu_b^*, P\left(f_g\right) = b$. The total redundant satisfying degree for all objectives is $rs = \sum_{g=1}^{4} rs_g$. The smaller the *rs* is, the closer the satisfying degree and expectation satisfying degree are, and the more consistent the SSS scheme is with the preferences of decision makers. By comparison, we can find that both (a) and (b) have higher μ_b^*, however in (b), γ is bigger and *rs* is the smallest, which indicates that the difference of importance between objectives in (b) is more obvious. (c) has the largest γ, indicating that the difference of importance between objectives is the most obvious, but μ_b^* is low, and *rs* is the largest.

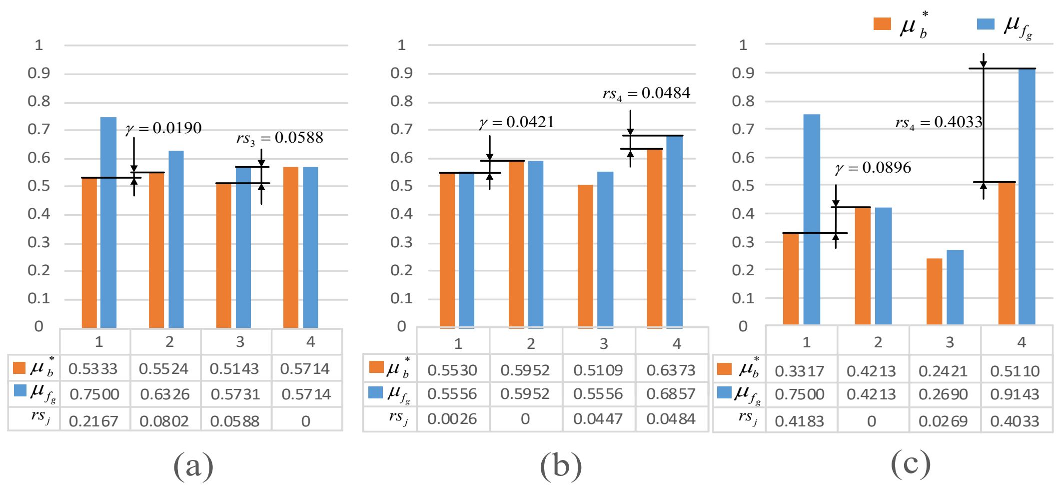

Figure 3. The satisfying degree and desirable satisfying degree corresponding to the best feasible SSS results found by (**a**) max-min method (**b**) two-phase method (**c**) weighted sum method.

Table 1 shows the optimal results found by the two-phase method with different $\Delta\delta$. It can be seen that the smaller $\Delta\delta$, the larger γ is, which also means more search space.

Table 1. The optimization results found by the two-phase method with different $\Delta\delta$ for example 4.1.

$\Delta\delta$	γ	Desirable Satisfying Degree	Satisfying Degree	rs
0.95	0.011	[0.547 0.558 0.536 0.569]	[0.731 0.558] 0.598 0.718]	0.396
0.9	0.058	[0.577 0.635 0.519 0.692]	[0.577 0.681 0.602 0.718]	0.156
0.85	0.059	[0.575 0.634 0.517 0.692]	[0.577 0.694 0.555 0.692]	0.101
0.8	0.086	[0.547 0.633 0.462 0.718]	[0.577 0.715 0.469 0.718]	0.120
0.75	0.086	[0.547 0.633 0.462 0.718]	[0.615 0.704 0.608 0.718]	0.286
0.7	0.089	[0.514 0.603 0.425 0.692]	[0.577 0.694 0.555 0.692]	0.283

In this example, it can be seen that different SSS methods result in different satisfying degree μ_{f_g} and desirable satisfying degree μ_b^*. It is necessary to investigate the performance of different SSS methods on multiple objectives in different dataset sizes. More results are presented in the next subsection.

4.2. Computational Experiments

4.2.1. Data Generation

Problems of different sizes were generated to compare the three methods. The main difference is that these problems have a different number of tasks, subtasks, and services. The number of services includes 6, 12, and 18, the number of tasks includes 5, 10, and 15, and the number of subtasks includes 8, 10, and 12. Therefore, there are $3^3 = 27$ possible combinations. We chose 9 of the 27 to test these three methods, i.e., 6s5t8st, 6s10t10st, 6s15t12st, 12s5t10st, 12s10t12st, 12s15t12st, 18s5t8st, 18s10t8st, 18s15t10st. For each combination, one group of datasets is generated, in which the relative important of the objectives, the service eligibility to fulfill each subtask. The available services and related information for each subtask are randomly generated in a uniform distribution. At the same time, two cases, $\Delta\delta = 0.9$, and $\Delta\delta = 0.7$, are set to test the effect of different $\Delta\delta$ value on the final solution.

Table 2 summarizes the ranges of *st* (service time), *sc* (service cost), *q* (quality), *ec* (environmental cost), *we* (weight) and *d* (distance) based on which the datasets are generated.

Table 2. The ranges of parameters.

st	*sc*	*q*	*ec*	*we*	*d*
[1,10]	[40,80]	[0.5,1]	[5,15]	[10,30]	[1,500]

4.2.2. Define GA in Full Term

The genetic algorithm used in the experiment contains 50 individuals per generation, up to 100 iterations, with a crossover probability of 0.8 and a mutation probability of 0.1. The selection of these parameters was based on published studies and preliminary tests were performed [29–31]. Here we no longer adjust the parameters because it takes a lot of time. However, by adjusting the parameters, it is entirely possible to find better results than in this paper. In order to get the statistical results of the optimal solution, each method runs 10 times for each selected problem. The experiments are all on Matlab. The operating environment of the computer is Intel (R) Core (TM) i5-7200U CPU @ 2.50Ghz, 16 GB RAM and Windows 10 operating system.

4.2.3. Test Results

The following performance indicators are considered:

- μ_4^*: The desirable satisfying degree of the least important objectives, representing the overall optimization level of all objectives.
- γ: Parameter of importance, which means the difference of satisfying degree between objectives with different importance.
- *rs*: Redundant satisfying degree, which means the difference between an actual satisfying degree and a corresponding desirable satisfying degree.

The results from ten runs are recorded, and the average of μ_4^*, γ, *rs*, CPU time are calculated separately. The results are shown in Table 3, and the following can be observed.

Compared with the weighted sum method, the max-min method and two-phase method can always find a larger μ_4^*. In most cases, μ_4^* in the two-phase method is slightly larger than the max-min method, but the difference is not very stable.

The γ in the two-phase method is always not less than the max-min method, and most of the cases are greater than. The weighted sum method can find a larger γ than the two-phase method, but there are some exceptions, especially when $\Delta\delta = 0.7$.

rs in the two-phase method is generally smaller than the max-min method, especially when the task size is larger, but there are exceptions when the task size is small. *rs* in the weighted sum method is much larger than the other two methods.

As the size of tasks, subtasks and services increases, the time spent by the three methods also increases, and the size of tasks and subtasks has a greater impact than the number of services.

Compared to $\Delta\delta = 0.9$, when $\Delta\delta = 0.7$, all three methods can find lower μ_4^*, higher γ. The CPU time consumed by the two-phase method increases significantly as $\Delta\delta$ becomes smaller, while it does not change too much under the other two methods.

In conclusion, the two-phase method outperforms the other two methods in finding the desirable satisfying degrees, which reflect the linguistic preference of decision makers. The two-phase method can not only achieve the collaborative optimization of all desirable satisfying degree, but also ensure the difference between them. In addition, the difference between a desirable satisfying degree and an actual satisfying degree is relatively small. The two-phase method has a good performance, but it takes more time, especially $\Delta\delta$ is relatively low. It should be noted that by adjusting the weight value, the weighted sum method can also get more satisfactory results for decision makers, but the inconsistency between the objective weight and the actual satisfying degree still exists. In addition, it is difficult for decision makers to translate linguistic preferences into precise weight values. Hence, for the SSS problem in cloud manufacturing, when the preference information of DM is expressed by linguistic terms, the two-phase method provides a better choice.

Table 3. Effects of different scheduling methods.

Dataset	$\Delta\delta$	Max-min				Two-phase				Weighted Sum			
		μ_4^*	γ	rs	CPU(s)	μ_4^*	γ	rs	CPU(s)	μ_4^*	γ	rs	CPU(s)
6s5t8st	0.9	0.476	0.030	0.115	0.881	0.491	0.032	0.129	1.252	0.396	0.038	0.664	0.901
	0.7	0.386	0.086	0.147	0.931	0.377	0.093	0.159	11.938	0.283	0.101	0.703	0.941
6s10t10st	0.9	0.436	0.028	0.173	2.034	0.426	0.032	0.097	2.384	0.293	0.031	0.889	2.041
	0.7	0.341	0.079	0.254	2.108	0.336	0.101	0.135	15.25	0.265	0.093	0.717	2.066
6s15t12st	0.9	0.439	0.034	0.169	3.389	0.446	0.044	0.144	3.668	0.197	0.211	0.577	3.558
	0.7	0.345	0.082	0.386	3.340	0.336	0.116	0.212	18.083	0.119	0.267	0.638	3.538
12s5t10st	0.9	0.512	0.020	0.203	1.060	0.524	0.025	0.108	1.591	0.281	0.114	0.496	1.100
	0.7	0.412	0.062	0.301	1.087	0.408	0.071	0.226	19.172	0.243	0.155	0.416	1.063
12s10t12st	0.9	0.452	0.022	0.189	2.289	0.439	0.022	0.152	2.524	0.283	0.030	0.985	2.285
	0.7	0.329	0.054	0.259	2.533	0.321	0.068	0.152	9.450	0.136	0.050	1.241	2.492
12s15t12st	0.9	0.453	0.029	0.142	3.59	0.440	0.037	0.128	4.323	0.192	0.110	0.688	3.621
	0.7	0.355	0.064	0.334	4.018	0.345	0.100	0.133	32.194	0.153	0.168	0.503	3.580
18s5t8st	0.9	0.430	0.052	0.155	0.950	0.443	0.075	0.165	1.381	0.325	0.125	0.694	1.023
	0.7	0.356	0.155	0.235	1.049	0.374	0.189	0.167	13.606	0.250	0.177	1.009	0.940
18s10t8st	0.9	0.408	0.019	0.142	1.655	0.408	0.019	0.111	1.882	0.213	0.012	1.267	1.637
	0.7	0.325	0.051	0.166	1.666	0.340	0.061	0.099	9.125	0.203	0.044	1.036	1.702
18s15t10st	0.9	0.480	0.027	0.159	3.190	0.487	0.036	0.119	3.492	0.297	0.070	0.950	3.260
	0.7	0.350	0.065	0.270	3.532	0.368	0.080	0.144	13.430	0.217	0.085	0.842	3.242

4.3. Performance Stability of Different Scheduling Schemes

To verify the stability of the three methods, larger scale services, tasks, and subtasks were tested, and finally, the results of different numbers of relative importance levels were observed.

4.3.1. Different Scales of Services and Tasks/Subtasks

Here, we have greatly increased the number of services, from the previous 6, 12, 18 to 60 and 600. The ranges of service time, service cost, quality, environmental cost, weight and distance follows data in Table 2. The range of parameters here and the definition of GA, is the same as in Section 4.2.

The results are shown in Table 4, the max-min method does not perform well than the two-phase method both in terms of increasing γ or decreasing rs, especially when $\Delta\delta$ is small. As the number of tasks in the problem increases, all three methods need to consume more CPU time. However, only the increase in the number of services does not necessarily lead to an increase in CPU time. Specifically, the weighted sum method requires relatively little computation time, and the advantage of the two-phase method is reflected in the efficiency of finding a satisfactory solution.

Table 4. Performance stability of different methods for different scales.

Dataset	$\Delta\delta$	Max-min				Two-phase				Weighted Sum			
		μ_4^*	γ	rs	CPU(s)	μ_4^*	γ	rs	CPU(s)	μ_4^*	γ	rs	CPU(s)
60s15t15st	0.9	0.472	0.057	0.210	4.729	0.470	0.090	0.143	5.317	0.401	0.147	0.554	4.611
	0.7	0.362	0.161	0.400	4.229	0.364	0.206	0.231	44.98	0.288	0.197	0.886	5.093
600s15t15st	0.9	0.405	0.015	0.247	4.746	0.412	0.019	0.180	4.939	0.180	0.018	1.506	4.257
	0.7	0.346	0.052	0.314	5.014	0.341	0.057	0.242	16.57	0.084	0.031	1.690	4.416
600s15t30st	0.9	0.460	0.024	0.169	10.61	0.437	0.023	0.183	10.35	0.285	0.032	0.925	9.817
	0.7	0.331	0.050	0.343	9.724	0.345	0.070	0.246	42.96	0.202	0.084	0.841	9.318
600s30t15st	0.9	0.463	0.023	0.245	10.20	0.444	0.026	0.200	12.47	0.233	0.153	0.566	10.90
	0.7	0.325	0.051	0.166	1.666	0.340	0.061	0.099	9.125	0.203	0.044	1.036	1.702
600s50t50st	0.9	0.423	0.019	0.131	112.7	0.401	0.018	0.131	114.1	0.281	0.024	0.844	112.2
	0.7	0.302	0.045	0.222	113.4	0.313	0.063	0.124	246.5	0.157	0.060	0.830	118.7

4.3.2. Different Numbers of Relative Importance Levels

In the above-mentioned experiments, we considered four levels of relative importance, i.e., $b = 1, 2, 3, 4$. In this section, the number of relative importance levels changed from three to seven. In order to test the performance of different methods, we chose combination 12s10t12st to conduct the experiment. All other parameters are the same as Section 4.3.1, and the results obtained by the three methods are shown in Table 5. It can be discovered that: the two-phase method is obviously superior to the other two methods when the number of levels is three, four, and five. However, for the situation that the number of level equals six and seven, the rs increases obviously in the two-phase method, even though the solution still keeps a higher μ_4^* and a larger γ. It indicates that the deviation between expectation satisfaction and actual satisfaction increases significantly. This is mainly because the number of levels is relatively large, resulting in the satisfying degree of the most important objective and the least important objective being too different. Therefore, the two-phase method is suitable for multi-objective optimization problems which need to achieve the optimization of all objectives, while maximizing the difference in optimization effects among objectives of different importance.

Table 5. Performance stability of different methods for different levels.

Level	$\Delta\delta$	Max-min				Two-phase				Weighted Sum			
		μ_4^*	γ	rs	CPU(s)	μ_4^*	γ	rs	CPU(s)	μ_4^*	γ	rs	CPU(s)
3	0.9	0.474	0.035	0.158	2.416	0.465	0.046	0.163	3.222	0.250	0.220	0.371	2.369
	0.7	0.365	0.089	0.351	2.566	0.362	0.121	0.164	14.992	0.185	0.278	0.477	2.352
4	0.9	0.450	0.018	0.167	2.402	0.487	0.027	0.124	2.724	0.249	0.023	1.129	2.368
	0.7	0.361	0.054	0.276	2.676	0.350	0.070	0.142	15.191	0.145	0.057	1.205	2.542
5	0.9	0.446	0.014	0.117	2.247	0.452	0.018	0.121	2.581	0.193	0.010	1.401	2.240
	0.7	0.353	0.043	0.187	2.333	0.356	0.047	0.118	15.812	0.175	0.034	1.185	2.447
6	0.9	0.464	0.016	0.164	2.472	0.449	0.027	0.567	3.214	0.142	0.052	1.171	2.426
	0.7	0.358	0.042	0.350	2.551	0.349	0.062	0.545	20.836	0.125	0.107	0.784	2.301
7	0.9	0.461	0.012	0.189	2.381	0.470	0.018	0.572	2.892	0.135	0.035	1.097	2.375
	0.7	0.356	0.032	0.350	2.592	0.356	0.050	0.510	22.373	0.092	0.048	1.114	2.375

5. Conclusions

For the multi-objective service selection and scheduling (SSS) problem with linguistic preference in cloud manufacturing (CMfg), a novel two-phase interactive optimization method is proposed in this paper. Whether the decision maker (DM) is satisfied is the final choice criteria. In the proposed method, the order of the desirable satisfying degrees is introduced and used to express the vague relative importance among the objectives showed by linguistic information. Next, the original problem is decomposed into two sub-problems and solved sequentially. The first-phase model aims to maximize overall satisfaction and achieves that all objectives are as close to the ideal value as possible. Then, in the second-phase model, the objective value of the first phase is taken as the constraint of the second phase and gradually relaxed until a solution that satisfies the DM is found. In the problem-solving process, the second phase is the most important, and if the DM is not satisfied, the next relaxation parameter will be given. Through these two phases, the optimization of all objectives and difference control of satisfying degrees are both achieved. The current method is easy to be implemented and applied. In order to apply the current method to cloud manufacturing, what needs to be done is to identify the number of relative importance levels and DM's linguistic preference for each objective. The benefit of quantifying the difference between satisfying degrees is that the DM can control the objective value more accurately in the process of decision-making, while avoiding giving the exact weight value before decision-making.

In the future, we can conduct deeper research from multiple directions. First, it is desirable to investigate other types of preference information, such as different priority of objectives. Secondly, the influence of adjustment of preference information in the decision-making process is also worth exploring. Furthermore, more advanced algorithms can be explored to reduce computation time.

Author Contributions: W.H., G.J., H.Z., J.K. conceived and designed the research; W.H. constructed the model and drafted the manuscript; H.Z. wrote the introduction and conclusion. J.K. designed the computational experiments; G.J. revised the manuscript.

Funding: The research was funded by the National Natural Science Foundation of China (Grant No.71772010), and the Technical Research Foundation (Grant No. JSZL2016601A004).

Acknowledgments: The authors would like to thank the reviewers and the editor for their insightful comments, which helped improve the quality of the paper significantly.

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

Appendix A

Table A1 presents the information about tasks and services, such as: alternative services, time, cost, quality, environmental cost, weight of product. Let's take the subtask 4 of task 2 as an example. This subtask could be executed on service $SE_{1,1}$, $SE_{1,2}$, $SE_{2,1}$, $SE_{2,2}$. If subtask $ST_{2,4}$ is assigned to service $SE_{2,1}$, then $st_{2,4}^{2,1} = 2$, $sc_{2,4}^{2,1} = 74$, $q_{2,4}^{2,1} = 0.92$, $ec_{2,4}^{2,1} = 15$, $we_{2,4}^{2,1} = 14$.

Table A1. Task and service information.

T_j \ $ST_{j,k}$		$ST_{j,1}$				$ST_{j,2}$				$ST_{j,3}$				$ST_{j,4}$			
T_3	$SE_{i,h}$	[(1,1), (1,2), (2,1), (2,2)]				[(1,1), (1,2), (2,1), (2,2)]				[(1,1), (1,2), (2,1), (2,2)]				[(1,1), (1,2), (2,1), (2,2)]			
	st	9	9	8	9	5	8	10	7	9	10	5	7	7	6	8	4
	sc	63	60	73	74	44	44	69	77	59	63	71	64	72	52	61	42
	q	0.97	0.94	0.67	0.72	0.97	0.97	0.86	0.85	0.97	0.68	0.71	0.76	0.86	0.90	0.70	0.62
	ec	6	14	14	14	13	7	8	6	6	15	8	6	6	7	15	5
	we	22	18	22	19	22	18	22	21	24	22	19	23	21	19	25	17
T_2	$SE_{i,h}$	[(1,1), (1,2), (2,1), (2,2)]				[(1,1), (1,2), (2,1), (2,2)]				[(1,1), (1,2), (2,1), (2,2)]				[(1,1), (1,2), (2,1), (2,2)]			
	st	9	8	9	4	1	4	2	2	4	8	6	7	3	1	2	4
	sc	44	78	75	55	73	66	78	41	63	71	42	80	54	69	74	68
	q	0.91	0.61	0.63	0.90	0.62	0.52	0.58	0.50	0.97	0.50	0.82	0.95	0.72	0.67	0.92	0.62
	ec	13	12	13	10	12	14	11	14	8	5	12	12	10	11	15	8
	we	24	19	23	21	23	26	26	26	23	19	25	26	20	21	14	12
T_3	$SE_{i,h}$	[(1,1), (1,2), (2,1), (2,2)]				[(1,1), (1,2), (2,1), (2,2)]				[(1,1), (1,2), (2,1), (2,2)]				[(1,1), (1,2), (2,1), (2,2)]			
	st	4	2	3	5	3	7	3	3	6	4	7	6	5	8	8	8
	sc	58	76	45	55	50	75	67	55	58	60	75	76	54	69	72	49
	q	0.56	0.65	0.68	0.95	0.64	0.53	0.63	0.68	1.00	0.89	0.91	0.74	0.89	0.71	0.83	0.82
	ec	10	14	5	5	6	12	7	12	7	8	12	11	14	8	13	9
	we	15	19	15	15	17	17	14	12	28	23	22	22	18	20	20	17

The distance between enterprises is shown in Table A2. For example, $d_{1,2} = 222$. The logistics time parameter $\alpha = 0.08$, the logistics cost parameter $\beta = 0.005$.

Table A2. Geographical distance $d_{i,i'}$ between enterprises.

Enterprise	E_1	E_2
E_1	0	222
E_2	222	0

References

1. Li, B.; Zhang, L.; Wang, S.; Tao, F.; Cao, J.; Jiang, X.; Song, X.; Cai, X. Cloud manufacturing: A new service-oriented networked manufacturing model. *Comput. Integr. Manuf. Syst.* **2010**, *16*, 1–7. (In Chinese)
2. He, W.; Xu, L. A state-of-the-art survey of cloud manufacturing. *Int. J. Comput. Integr. Manuf.* **2015**, *28*, 239–250. [CrossRef]
3. Chen, J.; Huang, G.Q.; Wang, J.; Yang, C. A cooperative approach to service booking and scheduling in cloud manufacturing. *Eur. J. Oper. Res.* **2019**, *273*, 861–873. [CrossRef]
4. Tao, F.; Zhang, L.; Liu, Y.; Cheng, Y.; Wang, L.; Xu, X. Manufacturing Service Management in Cloud Manufacturing: Overview and Future Research Directions. *J. Manuf. Sci. Eng.* **2015**, *137*, 040912. [CrossRef]
5. Tao, F.; LaiLi, Y.; Xu, L.; Zhang, L. FC-PACO-RM: A parallel method for service composition optimal-selection in cloud manufacturing system. *IEEE Trans. Ind. Inform.* **2013**, *9*, 2023–2033. [CrossRef]
6. Xu, X. From cloud computing to cloud manufacturing. *Robot. Comput.-Integr. Manuf.* **2012**, *28*, 75–86. [CrossRef]
7. Zhang, Y.; Zhang, G.; Liu, Y.; Hu, D. Research on services encapsulation and virtualization access model of machine for cloud manufacturing. *J. Intell. Manuf.* **2017**, *28*, 1109–1123. [CrossRef]
8. Cheng, Y.; Tao, F.; Zhao, D.; Zhang, L. Modeling of manufacturing service supply-demand matching hypernetwork in service-oriented manufacturing systems. *Robot. Comput.-Integr. Manuf.* **2017**, *45*, 59–72. [CrossRef]
9. Shen, X.; Yao, X. Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems. *Inform. Sci.* **2015**, *298*, 198–224. [CrossRef]
10. Wang, S.; Guo, L.; Kang, L.; Li, C.; Li, X.; Stephane, Y.M. Research on selection strategy of machining equipment in cloud manufacturing. *Int. J. Adv. Manuf. Technol.* **2014**, *71*, 1549–1563. [CrossRef]
11. Liu, W.; Liu, B.; Sun, D.; Li, Y.; Ma, G. Study on multi-task oriented services composition and optimisation with the "Multi-Composition for Each Task' pattern in cloud manufacturing systems. *Int. J. Comput. Integr. Manuf.* **2013**, *26*, 786–805. [CrossRef]
12. Tao, F.; Cheng, J.; Cheng, Y.; Gu, S.; Zheng, T.; Yang, H. SDMSim: A manufacturing service supply-demand matching simulator under cloud environment. *Robot. Comput.-Integr. Manuf.* **2017**, *45*, 34–46. [CrossRef]
13. Chen, T. Strengthening the competitiveness and sustainability of a semiconductor manufacturer with cloud manufacturing. *Sustainability* **2014**, *6*, 251–266. [CrossRef]
14. Wu, D.; Greer, M.J.; Rosen, D.W.; Schaefer, D. Cloud manufacturing: Strategic vision and state-of-the-art. *J. Manuf. Syst.* **2013**, *32*, 564–579. [CrossRef]
15. Liu, Y.; Xu, X.; Zhang, L.; Wang, L.; Zhong, R.Y. Workload-based multi-task scheduling in cloud manufacturing. *Robot. Comput.-Integr. Manuf.* **2017**, *45*, 3–20. [CrossRef]
16. Akbaripour, H.; Houshmand, M.; van Woensel, T.; Mutlu, N. Cloud manufacturing service selection optimization and scheduling with transportation considerations: Mixed-integer programming models. *Int. J. Adv. Manuf. Technol.* **2018**, *95*, 43–70. [CrossRef]
17. Cheng, Y.; Tao, F.; Liu, Y.; Zhao, D.; Zhang, L.; Xu, L. Energy-aware resource service scheduling based on utility evaluation in cloud manufacturing system. *Proc. Inst. Mech. Eng. B J. Eng.* **2013**, *227*, 1901–1915. [CrossRef]
18. Huang, B.; Li, C.; Tao, F. A chaos control optimal algorithm for QoS-based service composition selection in cloud manufacturing system. *Enterp. Inf. Syst. UK* **2014**, *8*, 445–463. [CrossRef]
19. Xiang, F.; Hu, Y.; Yu, Y.; Wu, H. QoS and energy consumption aware service composition and optimal-selection based on Pareto group leader algorithm in cloud manufacturing system. *Cent. Eur. J. Oper. Res.* **2014**, *22*, 663–685. [CrossRef]
20. Li, C.; Wang, S.; Kang, L.; Guo, L.; Cao, Y. Trust evaluation model of cloud manufacturing service platform. *Int. J. Adv. Manuf. Technol.* **2014**, *75*, 489–501. [CrossRef]
21. Lartigau, J.; Xu, X.; Nie, L.; Zhan, D. Cloud manufacturing service composition based on QoS with geo-perspective transportation using an improved Artificial Bee Colony optimisation algorithm. *Int. J. Prod. Res.* **2015**, *53*, 4380–4404. [CrossRef]
22. Ahn, G.; Park, Y.; Hur, S. The dynamic enterprise network composition algorithm for efficient operation in cloud manufacturing. *Sustainability* **2016**, *8*, 1239. [CrossRef]

23. Tao, F.; Zhao, D.; Hu, Y.; Zhou, Z. Correlation-aware resource service composition and optimal-selection in manufacturing grid. *Eur. J. Oper. Res.* **2010**, *201*, 129–143. [CrossRef]
24. Wu, Z.; Liu, T.; Gao, Z.; Cao, Y.; Yang, J. Tolerance design with multiple resource suppliers on cloud-manufacturing platform. *Int. J. Adv. Manuf. Technol.* **2016**, *84*, 335–346. [CrossRef]
25. Cao, Y.; Wang, S.; Kang, L.; Gao, Y. A TQCS-based service selection and scheduling strategy in cloud manufacturing. *Int. J. Adv. Manuf. Technol.* **2016**, *82*, 235–251. [CrossRef]
26. Hwang, C.L.; Masud, A.S.M. *Multiple Objective Decision Making—Methods and Applications*; Springer: New York, NY, USA, 1979.
27. Narasimhan, R. GOAL PROGRAMMING IN A FUZZY ENVIRONMENT. *Decis. Sci.* **2010**, *11*, 325–336. [CrossRef]
28. Chen, L.H.; Tsai, F.C. Fuzzy goal programming with different importance and priorities. *Eur. J. Oper. Res.* **2001**, *133*, 548–556. [CrossRef]
29. Abdelmaguid, T.F.; Nassef, A.O.; Kamal, B.A.; Hassan, M.F. A hybrid GA/heuristic approach to the simultaneous scheduling of machines and automated guided vehicles. *Int. J. Prod. Res.* **2004**, *42*, 267–281. [CrossRef]
30. Liu, L.; Hu, R.; Hu, X.; Zhao, G.; Wang, S. A hybrid PSO-GA algorithm for job shop scheduling in machine tool production. *Int. J. Prod. Res.* **2015**, *53*, 5755–5781. [CrossRef]
31. Falzon, G.; Li, M. Enhancing genetic algorithms for dependent job scheduling in grid computing environments. *J. Supercomput.* **2012**, *62*, 290–314. [CrossRef]