systems

Article

Resource Scheduling Optimisation Study Considering Both Supply and Demand Sides of Services under Cloud Manufacturing

Qinglei Zhang [1], Ning Li [2,*], Jianguo Duan [1], Jiyun Qin [1] and Ying Zhou [1]

[1] Logistics Engineering College, Shanghai Maritime University, Shanghai 201306, China; qlzhang@shmtu.edu.cn (Q.Z.); jgduan@shmtu.edu.cn (J.D.); jyqin@shmtu.edu.cn (J.Q.); zhouying@shmtu.edu.cn (Y.Z.)

[2] Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 201306, China

* Correspondence: 202130510270@stu.shmtu.edu.cn; Tel.: +86-13181069428

Abstract: In cloud manufacturing environments, the scheduling of multi-user manufacturing tasks often fails to consider the impact of service supply on resource allocation. This study addresses this gap by proposing a bi-objective multi-user multi-task scheduling model aimed at simultaneously minimising workload and maximising customer satisfaction. To accurately capture customer satisfaction, a novel comprehensive rating index is introduced, integrating the actual completion cost, time, and processing quality against customer expectations. Furthermore, vehicle constraints are incorporated into the model to accommodate potential delays in transport vehicle availability, thereby enhancing its alignment with real-world manufacturing settings. The proposed mathematical model is solved using an improved three-stage genetic algorithm, which integrates the k-means algorithm and a real-time sequence scheduling strategy to optimise solution quality. Validation against alternative algorithms across various case scales demonstrates the efficacy of the approach in providing practical scheduling solutions for real-case scenarios.

Keywords: ISGA; workload; scheduling optimisation; satisfaction

check for updates

Citation: Zhang, Q.; Li, N.; Duan, J.; Qin, J.; Zhou, Y. Resource Scheduling Optimisation Study Considering Both Supply and Demand Sides of Services under Cloud Manufacturing. *Systems* **2024**, *12*, 133. https://doi.org/10.3390/systems12040133

Academic Editor: Frank Herrmann

Received: 22 March 2024
Revised: 7 April 2024
Accepted: 12 April 2024
Published: 15 April 2024

1. Introduction

Cloud manufacturing is a new customer-oriented manufacturing model based on cloud computing and advanced manufacturing technology. It uses a cloud computing platform to provide manufacturing enterprises with support for resource sharing, service collaboration, information integration, etc. It realises the dynamic organisation and scheduling of manufacturing resources through network connection and data interaction, so as to realise the flexible, efficient, and intelligent development of the manufacturing industry and thus better serve customers. The cloud manufacturing platform is composed of the cloud platform, users, and manufacturers. The selection and scheduling of cloud services from the perspective of the cloud platform, in addition to meeting user needs and improving user satisfaction, also need to consider the needs of the cloud service supply side, such as load balancing, in order to avoid the manufacturer's idle resources from being not reasonably utilised. Therefore, it is meaningful to study how to ensure load balancing on the service provider side while ensuring customer satisfaction.

In most studies, cloud manufacturing scheduling [1] primarily focuses on objectives, such as the time, cost, and service quality. However, in supply chain management, due to customer irrationality, satisfaction cannot be calculated through a simple linear relationship. Therefore, to better enhance customer satisfaction, this paper proposes a comprehensive evaluation index that compares the actual completion cost, time, and processing quality with customer expectations. Additionally, to safeguard the manufacturer's interests, we set a workload balancing objective to prevent resource overload or idleness among manufacturing service providers. Consequently, a novel multi-objective scheduling optimisation model is introduced, encompassing both service supply and demand sides, with objectives of minimising workload and maximising customer satisfaction. Notably, unlike traditional manufacturing models, the transportation aspect is crucial in cloud manufacturing due to the geographical dispersion of manufacturing service providers. In reality, service providers submit their remaining manufacturing capacity to the platform, and due to their own orders, transport vehicles may not be available immediately after sub-task completion, thereby affecting the processing time of subsequent sub-tasks. Hence, vehicle constraints are incorporated to better reflect real-world scenarios. Finally, an Improve 3-Stage Genetic Algorithm (ISGA) is proposed for this scheduling problem, which integrates the adaptive Pareto population method and k-means algorithm, and it adopts a real-time sequential scheduling strategy, which is capable of obtaining a better set of Pareto solutions compared with the other four algorithms.

The rest of the paper is structured as follows: Section 2 reviews the scheduling studies on cloud manufacturing, Section 3 describes, in detail, the multi-task multi-objective scheduling optimisation mathematical model proposed in this paper; the multi-objectives include satisfaction and workload. Section 4 details the improved three-stage genetic algorithm, and Section 5 presents the results of simulation experiments and validation of the effectiveness of the algorithm performance. Section 6 states the conclusions and discusses future research.

2. Literature Review

2.1. Cloud Manufacturing Scheduling Problems

In recent years, with the development of related theories and technologies, manufacturing service allocation and scheduling under cloud manufacturing have attracted the attention of a wide range of scholars [2], and in the face of large-scale manufacturing tasks from multiple users, how to optimise resource allocation and scheduling to satisfy users' needs has become a research hotspot. Zhang et al. [3] proposed a synergistic elementary service-group-based service composition (SESG-SC) approach, which allows multiple functionally equivalent elementary services to be freely combined into one SESG to jointly perform each subtask, improving the overall service quality. E.V. Goodarzi et al. [4] proposed a mathematical model of the manufacturer's portfolio based on game theory and proved that in the cooperative game model, the profit and the quality of service gained by the manufacturing service provider are greater than those in the non-cooperative game model. Qian et al. [5] designed a collaborative cloud platform to provide an optimal production schedule by considering the cost, time, quality, and energy issues in the manufacturing process. Omid Fatahi Valilai et al. [6] performed scheduling optimisation based on the customised requirements of dentistry by considering the completion time and total completion time as the optimisation objectives. Jian et al. [7] developed relevant mathematical optimisation models based on the dynamic implementation of the tasks in the cloud manufacturing to save the shop floor production time and cost. Liu et al. [8] developed a multi-objective mathematical model to minimise the total cost, carbon emission, and water resources from economic and environmental perspectives while meeting time, cost, and quality requirements. Mehdi Zeynivand et al. [9] used the concept of alternative process routing for task decomposition, which enhances the flexibility of the system to complete resource allocation, and task delivery at the lowest cost. Hu et al. [10] established a scheduling metric assessment based on five aspects, task loadability, task reliability, manufacturing efficiency, manufacturing resource richness, and IoT matching, and used the AHP method to determine the relative weights of each factor at each level to transform the multi-objective optimisation into a single-objective problem, improving timeliness of the cloud manufacturing platform. Li et al. [11] studied the task scheduling of industrial robots under different enterprises under cloud manufacturing and proposed three sub-task scheduling strategies based on the three optimisation objectives of balanced robot load, lowest total cost, and shortest total processing time, which were realised by genetic algorithms. Tong et al. [12], based on the irrational characteristics of the users, used the prospect theory in psychology to compute the user's satisfaction level for the tasks corresponding to advancement or delay satisfaction.

In previous studies, as shown in Table 1, most of the studies used the total completion time, total completion cost, and service quality as objectives, some studies used cost, time, and service quality as constraints, and some studies used a hybrid approach. Similar to the hybrid approach, we use customer satisfaction as an objective and time, cost, and processing quality as objective components and constraints. At the same time, we study the scheduling problem from two perspectives, the service demand side and the service supply side, and set two objectives, maximum satisfaction and workload balance.

Table 1. Summary of research objectives.

		Demand Side				**Supply Side**	
Author	**Year**	**Cost**	**Time**	**Quality of Service**	**Customer Satisfaction**	**Energy Consumption**	**Workloads**
Li et al. [11]	2017	√	√			√	
Akbaripour et al. [13]	2017	√	√	√			
Zhou et al. [14]	2017			√		√	
Delaram et al. [15]	2018	√					
Zhang et al. [16]	2019	√	√	√		√	
Zheng et al. [17]	2019	√		√		√	
Qian et al. [5]	2019	√	√	√			
Zhou et al. [18]	2020		√				
Li et al. [19]	2020	√	√	√			
Zhou et al. [20]	2021		√				
Wu et al. [21]	2022	√					
Tong et al. [12]	2022				√		

2.2. Scheduling Optimisation Algorithm Problem

Multi-objective scheduling problems in cloud manufacturing are mostly solved by obtaining Pareto solution sets, and meta-heuristic algorithms are one of the commonly used tools for optimisation problems. Helo et al. [22] built a cloud-based production scheduling system for sheet metal manufacturing and developed a genetic-algorithm-based scheduling application to serve distributed manufacturing lines in the form of cloud manufacturing. Gilseung Ahn [23] developed a mathematical model to minimise latency, cost, quality, and reliability and proposed a multi-objective genetic algorithm for real-time scheduling. Ghomi et al. [24] proposed a queuing network for the parallel scheduling of multi-tasks and solved the model using a particle swarm algorithm based on the processing time of the tasks. Li et al. [25] proposed a two-level multi-task scheduling model and found the optimal solution using an improved ant colony algorithm based on two different scheduling levels. Li et al. [19] considered the hierarchical relationship of sub-tasks within a task, taking time, cost, and service quality as the optimisation objectives. A multi-objective algorithm based on a swarm algorithm and multi-objective algorithm based on NSGA-II were designed to solve the scheduling problem. Yuan et al. [26] established an optimal scheduling model for dynamic service resources in the CMfg environment, with the objectives of time, cost, quality, and capacity, and proposed a genetic-ant optimisation fusion algorithm to solve the model. Shi et al. [27] established an optimisation model with the objectives of cost minimisation and time minimisation and proposed a new algorithm combining a bat algorithm and a metacellular automaton for a cloud manufacturing resource scheduling new algorithm, which transforms the continuous scheduling problem into a discrete problem to obtain the optimal solution. Wu et al. [21] studied the integrated cross-supplier order and logistics scheduling problem, and developed a strategy based on the improved mix-and-shuffle frog jumping algorithm to solve the model.

Therefore, an improved multi-objective genetic algorithm is designed in this paper to solve the multi-task scheduling problem. In order to improve the comprehensiveness of the scheduling solution in the initial population, a three-stage genetic algorithm is used to consider the service allocation and subtask ordering in stages, which improves the operation efficiency by gradually reducing the search space. Meanwhile, the K-means algorithm is combined with the real-time sequence scheduling strategy, which makes the quality of the solution improve during the iteration process.

3. Multi-Task Scheduling Optimisation Model for Supply and Demand Sides

3.1. Description of the Problem

The diagram of the cloud manufacturing model is shown in Figure 1; users continuously provide manufacturing task requirements to the cloud manufacturing platform through the Internet, and manufacturing service providers submit manufacturing services and capabilities to the cloud manufacturing platform system. The cloud manufacturing platform virtualises and services manufacturing resources, generates a set of optimal service scheduling solutions based on user task requirements, and finally sends them to each enterprise manufacturing workshop to complete the execution of specific tasks.

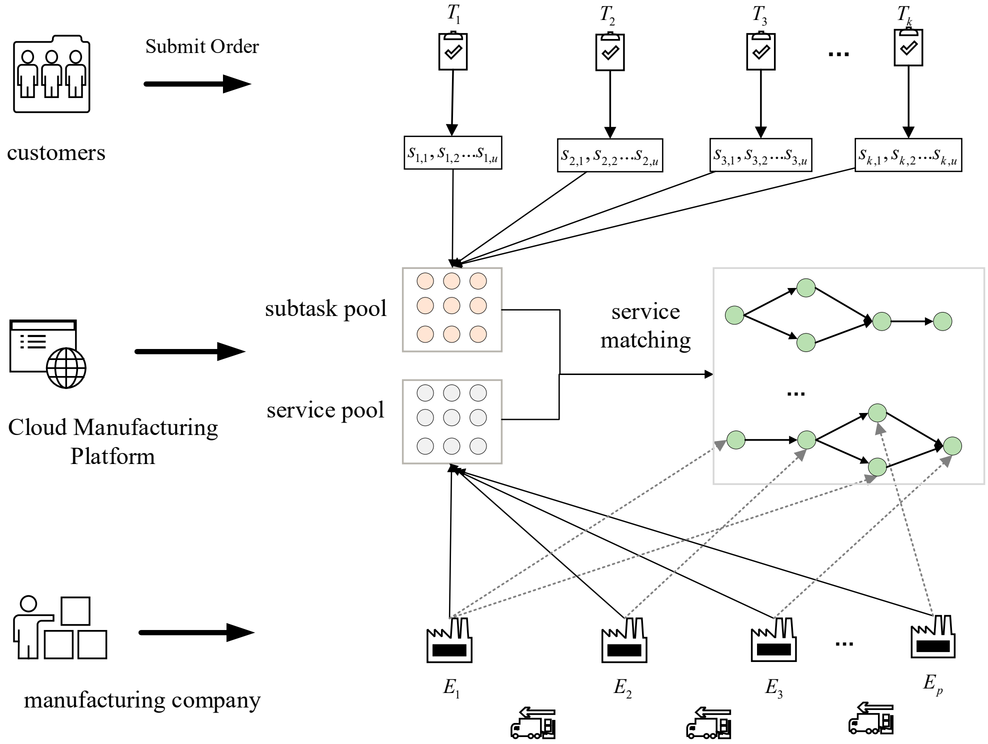
Submit Order
customers
T_1
T_2
T_3
...
T_k
$s_{1,1}, s_{1,2} \cdots s_{1,u}$
$s_{2,1}, s_{2,2} \cdots s_{2,u}$
$s_{3,1}, s_{3,2} \cdots s_{3,u}$
$s_{k,1}, s_{k,2} \cdots s_{k,u}$
subtask pool
service matching
service pool
...
Cloud Manufacturing Platform
manufacturing company
E_1
E_2
E_3
...
E_p

Figure 1. Cloud manufacturing flowchart.

Suppose that under the current cloud manufacturing platform consists of manufacturing service enterprises, denoted as $E = E_1, E_2, E_3 \ldots E_m$, which the enterprise E_m can provide i types of manufacturing services, which the type of services provided by the enterprise is denoted as $M_{m,r}$. The time, cost, and quality of processing vary from enterprise to enterprise for the same type of processing. Demand tasks from the cloud manufacturing platform user order decomposition, assuming different customers in the cloud manufacturing platform under the submission of single or multiple orders, the cloud platform to receive the allocation for the task processing.

The cloud platform receives the allocation for I tasks to process, denoted as $T = T_1, T_2, T_3, \ldots, T_I$, each task includes single or multiple sub-tasks, which may require one or more types of services, and T_i can be decomposed into j sub-tasks, denoted as $S_{i,1}$,$S_{i,2}$, $S_{i,3} \ldots$ $S_{i,j}$, where the corresponding attributes of each subtask are different and the attributes of the subtasks are denoted by the symbols $R_{i,j}$. The decomposed subtasks are performed by manufacturing resource providers in the CMfg system by providing manufacturing services. The following assumptions are made for the model.

(1) Tasks are all available and have the same priority at the beginning;
(2) That storage buffers are infinite;
(3) Manufacturers have unlimited manufacturing resources;
(4) A service provider can handle different types of subtasks, but can only service one subtask at a time;
(5) Different service providers process the same type of subtask with different costs and times;
(6) That the quality of service for completing different types of subtasks is different;
(7) That all subtasks cannot be interrupted once processing has begun.

The relevant parameters involved in the mathematical model are as shown in Table 2:

Table 2. Related symbols and meanings.

Notation	Hidden Meaning
T_i	Task i = 1, 2, 3. . .I
I	Total number of tasks
J_{T_i}	Number of subtasks in task i
$ST_{i,j}$	Subtask j of task i,j = 1, 2, 3. . .J
$R_{i,j}$	Attributes of the jth subtask of task i
E_m	Manufacturer m = 1, 2, 3. . .M
$M_{m,r}$	Attributes of services provided by manufacturer E_m
$PC_{i,j}^m$	Processing cost of sub-task j of task i at manufacturer m
$SC_{i,j}^m$	Installation cost of subtask j of task i at manufacturer m
$CC_{i,j}^m$	Manufacturing cost of sub-task j of task i at manufacturer m
$PT_{i,j}^m$	Processing time of task i's subtask j at manufacturer m
$ST_{i,j}^m$	Subtask j of task i's installation time at manufacturer m
$Q_{i,j}^m$	Quality of sub-task j of task i at manufacturer m
TC	Cost per unit distance travelled
TT	Time per unit distance travelled
$d_{m,n}$	Distance from manufacturer m to manufacturer n
$AST_{i,j}^m$	Start time of subtask j of task i at manufacturer m
$CT_{i,j}^m$	Completion time of sub-task j of task i at manufacturer m
AT_m	Available time at manufacturer m
$PST_{i,j}^m$	Preset start time of task i's subtask j at manufacturer m
$\alpha_{i,j}^m$	Whether sub-task j of task i is processed at manufacturer m or not
$C_{cont,i}$	Maximum acceptable cost for task i
$T_{cont,i}$	Latest acceptable time for task i
AB_p	Earliest available time for vehicle at E_p
$Q_{cont,i}$	Minimum acceptable product quality for task i
W	Total average workload
$\overline{RC}$	Weighted cost satisfaction
$\overline{RT}$	Weighted time satisfaction
$\overline{RS}$	Weighted quality satisfaction
w_i	Weights of different attributes in satisfaction
$\alpha_{i,j}^m$	1 if subtask $ST_{i,j}$ is processed by E_m; 0 otherwise

3.2. Multi-Task Scheduling Model

In this paper, the objective function of the multi-objective optimisation model is established as follows with the optimisation objectives of minimising the workload on the service supply side and maximising the satisfaction on the service demand side in the cloud manufacturing process:

Minimum workload:

$$f_1 = minW \tag{1}$$

Maximum demand-side satisfaction:

$$f_2 = \max SA \tag{2}$$

Based on the perspective of service providers, it aims to avoid the overloading of some service providers or idling of some service providers, thus improving the resource utilisation. Load balancing is controlled by calculating the sum of the absolute value of the difference between the working time and the average working time of each service provider, thus protecting the interests of the service providers. The calculation method is shown in Equation (3).

$$W = \sum_{m=1}^{M} \left|\sum_{i=1}^{I}\sum_{j=1}^{J} PT_{i,j}^{m} * \alpha_{i,j}^{m} - \frac{1}{M}\sum_{m=1}^{M}\sum_{i=1}^{I}\sum_{j=1}^{J} PT_{i,j}^{m} * \alpha_{i,j}^{m}\right| \tag{3}$$

Cloud manufacturing is the embodiment of the manufacturing-as-a-service concept, and customer satisfaction reflects the degree of customer satisfaction with the resource scheduling service provided by the cloud manufacturing platform, which is one of the key factors in evaluating the success of the cloud manufacturing model. In this paper, we investigate the comprehensive impact of the three dimensions of time, cost, and service quality on customer satisfaction SA.

Customer satisfaction with the total completion cost is expressed as the ratio of the actual completion cost to the maximum acceptable cost to the customer. Since the number of subtasks varies from task to task, for a more accurate representation of satisfaction, a weighted summation is used, with the size of the weight depending on the number of subtasks. When the value is smaller, it represents the lower actual total completion cost required, i.e., the higher the cost satisfaction of the customer. The formula is shown in Equation (4).

$$\overline{RC} = \sum_{i=1}^{I} \left(\left(\sum_{m=1}^{M}\sum_{j=1}^{J} \left(CC_{i,j} + LC_{(i,j),(i,j-1)}\right)/C_{cont,i}\right) * \frac{J_{T_i}}{\sum_{i=1}^{I} J_{T_i}}\right) \tag{4}$$

where the total cost consists of the manufacturing cost *CC* and the transport cost *LC*; the manufacturing cost *CC* consists of the installation cost *SC* and the processing cost *PC*, which are calculated in Equations (5) and (6).

$$CC_{i,j} = \left(SC_{i,j}^{m} + PC_{i,j}^{m}\right) * \alpha_{i,j}^{m} \tag{5}$$

$$LC_{(i,j),(i,j-1)} = TC * d_{m,n} * \alpha_{i,j}^{m} * \alpha_{i,j-1}^{n} \tag{6}$$

Customer satisfaction with the total completion time is expressed as the ratio of the actual total cost of completion to the customer's maximum acceptable cost. The smaller the value, the shorter the actual total completion time required, i.e., the higher the customer's time satisfaction. Total completion time includes the manufacturing time and logistics time. The calculation is shown in Equation (7).

$$\overline{RT} = \sum_{i=1}^{I} \left(\left(\sum_{m=1}^{M}\sum_{i=1}^{I}\sum_{j=1}^{J} \alpha_{i,j}^{m} * CT_{i,j}^{m}/T_{cont,i}\right) * \frac{J_{T_i}}{\sum_{i=1}^{I} J_{T_i}}\right) \tag{7}$$

In this paper, transport time *TT*, installation time *ST*, and processing time *PT* are considered, and the time required for different sub-tasks and selected service combinations is different. The specific calculations are shown in Equations (8)–(13).

$$PST_{i,j}^{m} = \begin{cases} ST_{i,j}^{m} & \text{No pilot mission} \\ ST_{i,j-1}^{m} + CT_{i,j-1} + LT_{(i,j),(i,j-1)} & \text{otherwise} \end{cases} \tag{8}$$

$$LT_{(i,j),(i,j-1)} = TT * d_{m,n} * \alpha_{i,j}^{m} * \alpha_{i,j-1}^{n} \tag{9}$$

where $PST_{i,j}^{m}$ represents the predicted start time of sub-task $ST_{i,j}$ when it is processed by service provider m, and $LT_{(i,j),(i,j-1)}$ is the time required to transport sub-task $ST_{i,j}$, which is processed by service provider n to service provider m. If service providers m and n are in the same company, then $LT_{(i,j),(i,j-1)} = 0$.

$$CT_{i,j}^{m} = AST_{i,j}^{m} + PT_{i,j}^{m} * \alpha_{i,j}^{m} \tag{10}$$

$$AST_{i,j}^{m} = max\left(PST_{i,j}^{m},\ AT_{m}\right) \tag{11}$$

$$AT_{m} = \max\left(max\left(CT_{i,j}^{m}\right), AA_{i,j-1}\right), \forall i,j \in \left\{\alpha_{i,j}^{m} = 1\right\} \tag{12}$$

Equation (10) represents the completion time of the subtask at service provider m. The actual start time of the subtask $AST_{i,j}^{m}$ depends on the actual start time of the task, while the actual start time of the subtask depends on the completion time of the previous subtask $CT_{i,j}^{m}$ and the current service provider's available time AT_{m}, the earliest time at which the service M can handle the current request for the subtask.

Since each manufacturing service provider in cloud manufacturing is geographically dispersed, the transport time required by the vehicles is non-negligible, so we consider two cases of the number of vehicles, which are finite and infinite. And if the manufacturing service provider completes the sub-tasks if they cannot be transported immediately, they are put into the warehouse for storage. When sub-task $S_{i,j}$ is processed at manufacturing service provider E_m, whether its subsequent sub-task $S_{i,j+1}$ can start immediately depends on the processing progress of another manufacturing service provider E_n, on the one hand, and on the transport time of sub-task $S_{i,j}$. And when the available time of the vehicle under the manufacturing service provider E_m is less than the completion time of the subtask, the subtask is put into the storage area until a free vehicle is available. As shown in Figure 2.

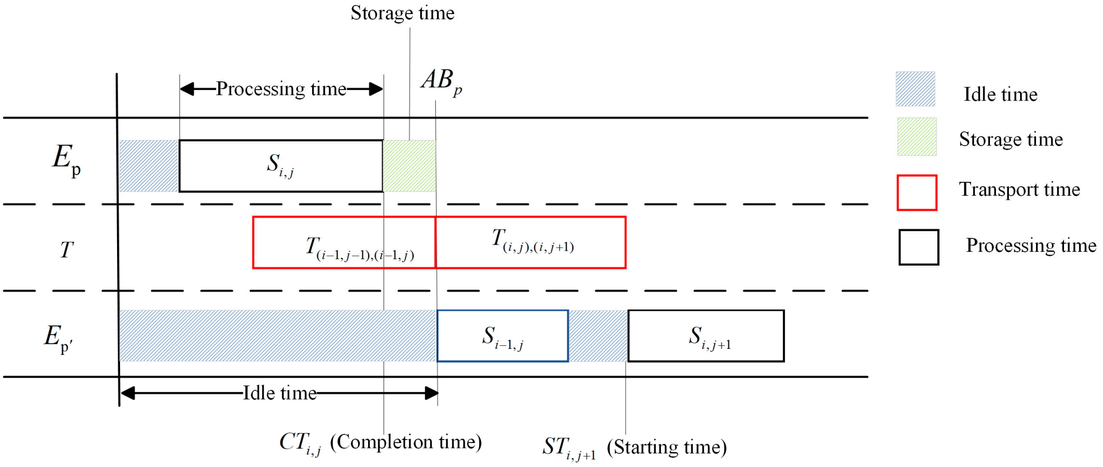

Figure 2. Scheduling instructions under vehicle restrictions.

Consequently, a priority transport strategy is considered to minimise transport time. This strategy includes the following: (1) transporting subtasks with services assigned by subsequent subtasks first if immediate initiation is not possible; (2) prioritising subtasks for transportation based on their completion time when either both sub-tasks can start immediately or neither can; (3) otherwise, placing them in the storage area to await transport.

$$\begin{cases} AA_{i,j} = CT_{i,j-1} + LT_{(i,j),(i,j-1)} \text{ Infinitely} \\ AA_{i,j} = \max\left(AB_p, CT_{i,j-1}\right) + LT_{(i,j),(i,j-1)} \text{ Limited} \end{cases} \tag{13}$$

Customer satisfaction with total service quality is expressed as the ratio of the actual total quality of completion to the minimum acceptable service to the customer. A higher value means a higher level of satisfaction. The formula is shown in (14).

$$\overline{RS} = \prod_{i=1}^{I}\left(\left(\prod_{m=1}^{M}\prod_{j=1}^{J}\frac{\alpha_{i,j}^{m} * Q_{i,j}^{m}}{Q_{i,j}^{m}}\right) * \frac{J_{T_i}}{\sum_{i=1}^{I} J_{T_i}}\right) \tag{14}$$

By assigning appropriate weights to different factors affecting customer satisfaction, a composite value of SA is calculated to better represent customer satisfaction. The calculation is shown in Equation (15).

$$SA = min\left(w_1 * \overline{RC} + w_2 * \overline{RT} + w_3 * \overline{RS}\right) \tag{15}$$

In order to better determine the share of the cost, time, and service quality in customer satisfaction, hierarchical analysis is used. Hierarchical analysis allows for the layering of problems. Depending on the nature of the problem and the objectives to be achieved, the problem is broken down into different components. The interactions and connections between the factors will form an ordered hierarchical model. The relative importance of the hierarchical factors in the model is then expressed quantitatively based on one's judgement of the objective reality. A comparative judgement matrix was constructed, and the weights for ranking the relative importance of each factor at each level were mathematically determined. Finally, the weights of the relative importance of each factor were comprehensively calculated as the basis for evaluation and selection.

The values of w_1, w_2, and w_3 were obtained through the calculation as 0.63, 0.26, and 0.11, and since the evaluation index of service quality is the bigger the better, the value of w_3 is negative for the purpose of unifying the indexes, and the values of w_3 in the final mathematical model are 0.63, 0.26, and -0.11.

3.3. Constraints

(1) The total cost of completion of the assignment cannot be higher than the maximum acceptable cost to the customer.

$$\sum_{m=1}^{M}\sum_{i=1}^{I}\sum_{j=1}^{J} CC_{i,j}^{m} * \alpha_{i,j}^{m} + LC_{(i,j),(i,j-1)} * \alpha_{i,j}^{m} * \alpha_{i,j-1}^{n} \le C_{cont,i} \tag{16}$$

(2) The total sub-task completion time under each task cannot exceed the latest customer acceptance time.

$$\max\left(CT_{i,j}^{m}\right) + \sum_{m=1}^{M}\sum_{i=1}^{I}\sum_{j=1}^{J} LC_{(i,j),(i,j-1)} * \alpha_{i,j}^{m} * \alpha_{i,j-1}^{n} \le T_{cont,i} \tag{17}$$

(3) A sub-task can be served by only one enterprise.

$$\sum_{m=1}^{M} \alpha_{i,j}^{m} = 1 \tag{18}$$

(4) The total cost of quality of the assignment cannot be lower than the maximum acceptable quality to the customer.

$$\sum_{m=1}^{M}\sum_{i=1}^{I}\sum_{j=1}^{J} \frac{Q_{i,j}^{m} * \alpha_{i,j}^{m}}{J_{T_i}} \ge Q_{cont,i} \tag{19}$$

(5) The service attributes that the manufacturing service provider can provide are matched one-to-one with the sub-task attributes in the customer's order.

$$R_{i,j} = M_{m,r}\ \mathrm{i,j,m}\epsilon\ \left(\alpha^{m}_{i,j} = 1\right) \tag{20}$$

4. Improved Three-Stage Genetic Algorithm Design Based on Clustering Algorithm

The model developed in this paper is designed to ensure workload balance while maximising customer satisfaction, so an improved three-stage genetic algorithm is proposed to solve the model. The framework of this research method consists of the Improve 3-Stage Genetic Algorithm (ISGA), which improves the operational efficiency by gradually reducing the search space.

In the first stage, only service allocation and cost constraints are considered. The initial population p is randomly generated based on the attribute consistency principle, and based on the k-means algorithm, the individuals are selected at different clustering centres, and then, the service vectors of each individual are cross-mutated to update the Pareto non-dominated solution set, as well as the non-dominated solution set, according to the Pareto method.

Before starting the second stage, the RTSS heuristic is used to initialise the subtask ordering vectors based on the final population obtained in the first stage, aiming to minimise the idle time based on the progress time of the services and the slack of the subtasks. When iteration η* is the maximum number of iteration times, the success rate of the population is lower than ζ in consecutive iterations, or when the number of iterations reaches θ_1*, the maximum number of iterations, the first stage is ended and the second stage is started. If both the parent and their descendants either satisfy or do not satisfy the time constraints, then the optimal solution is retained; otherwise, solutions that satisfy the time constraints are retained.

When iterating a maximum of η* maximum iterations, the population's success rate remains consistently below ζ in consecutive iterations, or when the number of iterations reaches θ_1 + θ_2* maximum iterations, the third stage begins. The parameters ζ, η, θ_1, and θ_2 are constant values between 0 and 1, with their specific values detailed in Section 5. Furthermore, in both the first and third stages, a Pareto adaptive method is employed to maintain population diversity and enhance the convergence speed by adjusting the population size based on the performance of descendants. The flow of the improved three-stage genetic algorithm is shown in Figure 3.

4.1. Adaptive Population Size

Pareto introduced the concept of non-dominated solutions for multi-objective problems in 1986. It is defined as follows: suppose for all objectives, f_1 is superior to f_2; in that case, we say that f_1 dominates f_2. If f_1 is not dominated by any other solution, then it is referred to as a non-dominated solution (also known as a non-dominated Pareto optimal solution). Achieving at least one better objective without making any other objective worse is known as Pareto improvement. The Pareto optimal state represents a condition where it is not possible to have more Pareto improvements. Cui et al. [28] proposed the Adaptive Method for Population Size (AMPS) and designed an external archive to store inferior individuals, and the individuals in the external archive still have a chance to enter the next generation. In this paper, we propose a Pareto adaptive population size method; the population size is adjusted according to the dynamic retention of individuals in the current population, which involves the following parameters: minimum population size SN_{min}, maximum population size SN_{max}, and the current population size SN. The population P consists of a Pareto non-dominated set M and a Pareto dominated set G. The set of Pareto non-dominated solutions includes all Pareto solutions and is continuously updated with each iteration.

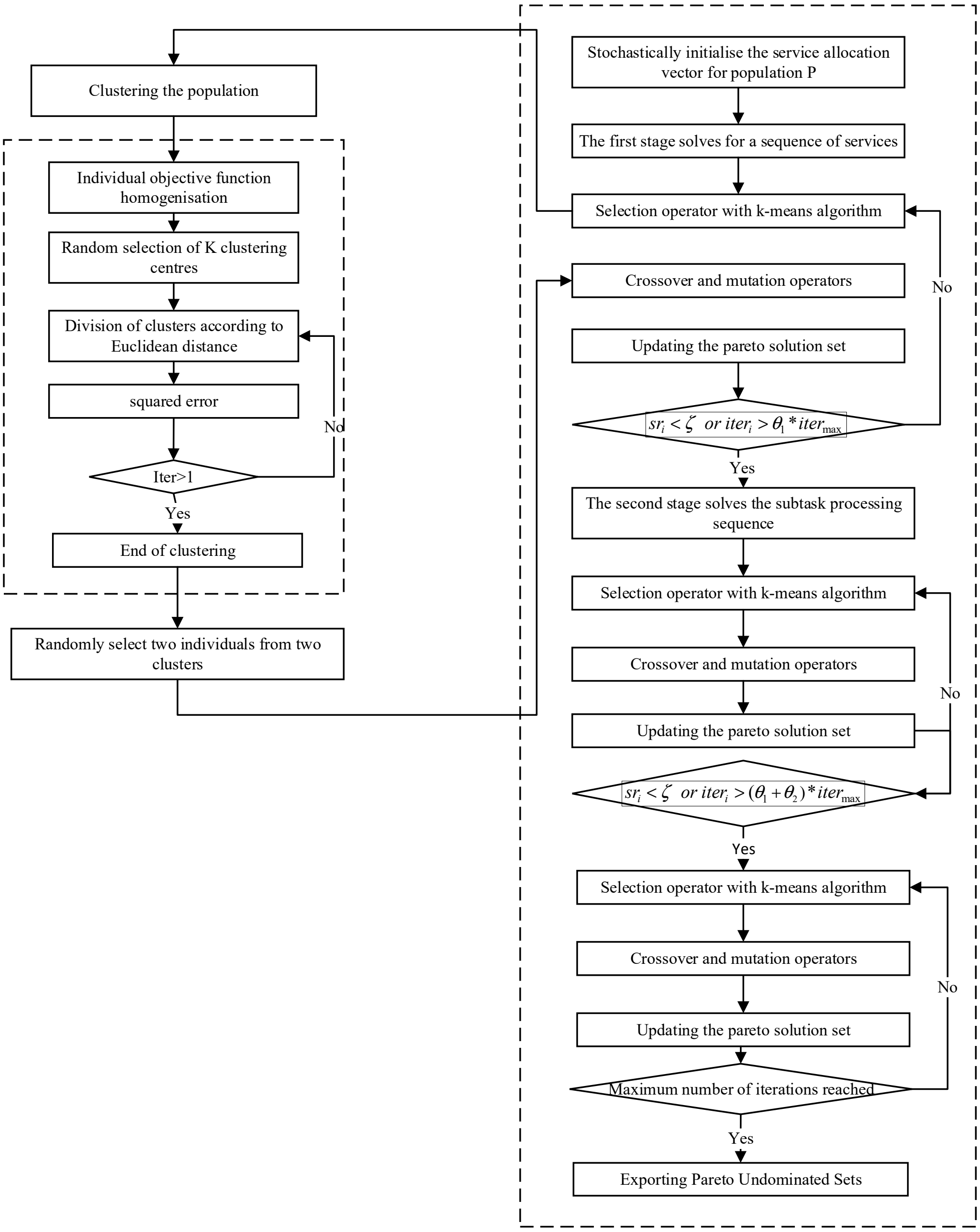
Clustering the population
Individual objective function homogenisation
Random selection of K clustering centres
Division of clusters according to Euclidean distance
squared error
Iter>1
No
Yes
End of clustering
Randomly select two individuals from two clusters
Stochastically initialise the service allocation vector for population P
The first stage solves for a sequence of services
Selection operator with k-means algorithm
Crossover and mutation operators
Updating the pareto solution set
$sr_i < \zeta$ or $iter_i > \theta_1 * iter_{\max}$
No
Yes
The second stage solves the subtask processing sequence
Selection operator with k-means algorithm
Crossover and mutation operators
Updating the pareto solution set
$sr_i < \zeta$ or $iter_i > (\theta_1 + \theta_2) * iter_{\max}$
No
Yes
Selection operator with k-means algorithm
Crossover and mutation operators
Updating the pareto solution set
Maximum number of iterations reached
No
Yes
Exporting Pareto Undominated Sets

Figure 3. Flowchart of the improved three-stage genetic algorithm.

In order to maintain the diversity and quality of the population, individuals that are dominant in their parents or offspring will be retained into the next generation. If the parents and offspring do not mutually dominate, the retention rule will change based on the population's success rate, calculated as $SR = SS - SN$, where SS represents the number of successful searches by individuals in the current population. Generally, if the offspring dominate their parents, the search for offspring is considered successful. If the population's success rate is high, the population contracts. The success rate of an individual is calculated using the formula

$$SR_i = SC_i - iter_i \tag{21}$$

where SC_i is the number of successful searches for individual X_i, and iter$_i$ is the number of iterations for X_i. If an individual X_i has a low success rate, X_i belonging to set G and its descendants X_i' are often not retained for the next generation; otherwise, X$_i$ will be retained. Conversely, if the population's success rate is low, the population tends to expand.

Non-dominated offspring are retained for the next generation. Of course, all population contraction or expansion is based on not exceeding the population size limit. The specific pseudocode is shown in Appendix A.

4.2. Genetic Algorithm

4.2.1. Modes of Coding

For the scheduling optimisation problem, service allocation and subtask sequencing are involved simultaneously. The service allocation problem is to select the appropriate manufacturing service for each subtask, and subtask sequencing is to confirm the processing order of the subtasks from the manufacturing service provider. Therefore, a two-tier coding approach is used. If there are M users, each with I tasks and each with J subtasks, the calculation of the sum of subtasks S can be performed according to Equation (22).

$$S = \sum_{m=1}^{M} \sum_{i=1}^{I} \sum_{j=1}^{J} st_{mnj} \tag{22}$$

The chromosome is encoded in an integer, and the length of the chromosome string is 2 × S. The first part indicates the service assignment, and the change service can process the corresponding subtasks. The second part indicates the processing order of all subtasks. For example, in MS = (2 3 1 2 1 1 3 3 1), the first '2' indicates that the first subtask of task 1 is assigned to the second manufacturing service provider for processing. The second layer is the subtask ordering, where each element represents a subtask of the corresponding task, and the nth occurrence of the same number indicates the nth subtask in the sequence of tasks. For example, for sequences = (1 1 2 1 3 3 2 3), the first '1' represents the first subtask of the first task. The second '1' represents the second subtask of the first task, and the length of the encoding is determined by the number of subtasks. The length of the encoding is determined by the subtask's. The encoding is 2312133111213323, The service sequence is $\{(T_{1,1}, MS_2), (T_{1,2}, MS_3), (T_{2,1}, MS_2), (T_{1,3}, MS_1), (T_{3,1}, MS_3), (T_{3,2}, MS_3), (T_{2,2}, MS_1),$ $(T_{3,3}, MS_1)\}$ which represents the scheduling scheme as shown in Figure 4.

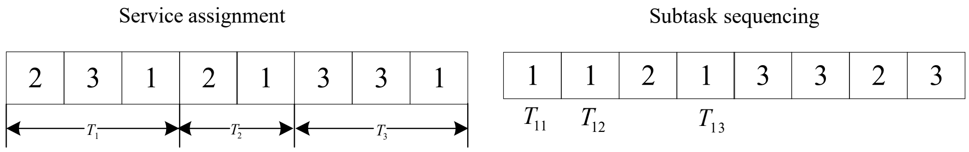

Figure 4. Modes of coding.

4.2.2. Binary Tournament Selection Method Based on k-Meas Algorithm

Binary tournaments usually randomly select two competitors in the population, and the winner goes to the next generation. However, since there is a probability of selecting similar individuals during the selection process, the k-mean algorithm is combined to improve the diversity of the population. To facilitate the calculation, the objective function value of each individual is standardised, k individuals are randomly selected as the clustering centre, divided into different clusters according to the Euclidean distance between each individual and the cluster, and the clustering centre is constantly updated according to the squared error until it reaches the satisfaction condition. Then, two clusters are randomly selected, and an individual with a minimum Pareto rank and larger crowding distance is selected from any two individuals in each cluster for the next step. The flow of the algorithm is as follows in Appendix B.

4.2.3. Cross-Variation

For service allocation sequences, a two-point crossover operator is used. Two different parents from the current population are selected as the basis for crossover, two crossover points are randomly selected on the chromosomes of the parents, and the segment between the crossover points is copied from one parent to the offspring individual, and the segment outside the crossover points from the other parent is copied to the offspring individual, as shown in Figure 5. Blue numbers indicate genes from parent 1 and red numbers indicate genes from parent 2 in Figure 5.

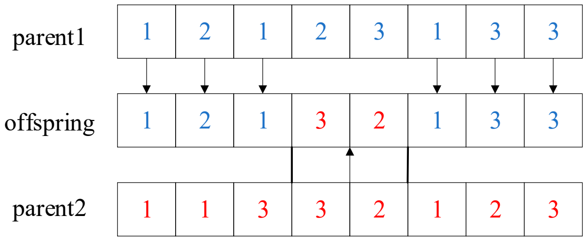

Figure 5. Two-point crossover.

For subtask ordering sequences, order-based priority-preserving crossover is used. Two different parent individuals from the current population were selected as the basis for crossover, and one or more crossover points were randomly selected in the aligned sequence of parent individuals. The gene segments between the crossover points were copied from one parent individual to the corresponding positions in the offspring, and the remaining blank positions were sequentially complemented by the remaining elements of the other parent individual, as shown in Figure 6. Blue numbers indicate genes from parent 1 and red numbers indicate genes from parent 2 in Figure 6.

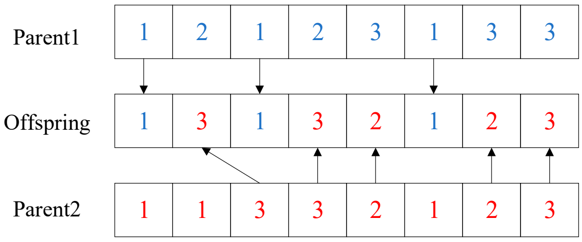

Figure 6. Order-based priority-preserving crossover.

For the service assignment sequence, a single point of mutation is used, where an individual from the current population is randomly selected as the subject of the mutation operation to be performed. A randomly selected mutation point in the chromosome of the selected individual, i.e., the location of the gene for which the mutation is to be performed, is re-scheduled to another service based on the mutation rate MR to produce a new offspring. If any infeasible offspring is generated, the available service closest to the currently unavailable service will be reselected and assigned to the corresponding subtask, as shown in Figure 7. Bolded text represents mutated genes and red text represents mutated genes in Figure 7.

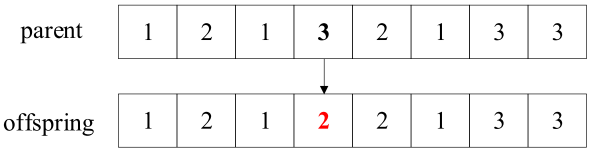

Figure 7. Single-point mutation.

For the subtask sequencing sequence, the mutation process will be carried out on the basis of the crossover operator by using the two-point mutation, where an individual is randomly selected from the current population as the object to carry out the mutation operation. Two mutation points, i.e., the gene positions where the mutation is to be performed, will be randomly selected in the chromosome of the selected individual and the two mutation points will be exchanged. The above mutation method avoids the generation of infeasible offspring, as shown in Figure 8. The blue and red numbers in Figure 8 represent the two different mutant genes that were selected.

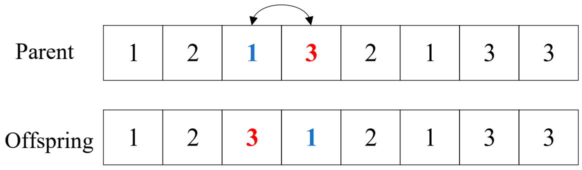

Figure 8. Two-point mutation.

4.3. Real-Time Task Sequence Scheduling

The purpose of real-time task sequence scheduling is to obtain an initial sub-task sequence based on the first-stage service allocation schedule. This sequence ensures that once the current sub-task processing order is determined, the subsequent sub-tasks become immediately available. During the scheduling process, newly available sub-tasks are sorted based on their slackness, and the sub-task sequencing sequence is updated until all sub-tasks are completed. If the selected sub-tasks have the same slackness, they are sorted randomly. The slackness of sub-tasks represents their urgency, and the calculation method is outlined in Formula (23):

$$L_{i,j} = TT_{i,j} - rt_i \tag{23}$$

where rt_i is the total time required to complete the remaining subtasks of the current task according to the given service assignment. To avoid time constraints that cannot be satisfied, all selected subtasks are sorted in ascending order based on their slackness values. Subtasks with the smallest slack value are scheduled first, and subtasks with the same slack value are randomly ordered. The pseudo-code is shown in Appendix C.

5. Case Study

5.1. Description of the Problem

In order to obtain effective parameter combinations and evaluate the performance of the T-SGA, a series of computational experiments was carried out based on four instances. The computer configuration used for these experiments was 8 GB RAM and an Intel(R) Core(TM) i7 CPU @ 1.80 GHz.

In order to test the efficiency of ISGA in multi-objective optimisation solving, six test instances were generated based on different numbers of tasks, sub-tasks, and manufacturing services, shown in Table 3. The ranges of different parameters in the test instances were collected from the company based on the stochastic nature of task execution in the CMfg system. The processing unit time and cost ranged from 10 min to 30 min and $10 to $40, and based on real applications, the unit time and cost were set to range from 10 min to 30 min and $10 to $30. Therefore, considering the confidentiality reasons, the values of these parameters were randomly generated from the scope of the survey. The distance between manufacturing service providers is taken between 100 and 400 km. The average unit logistics cost lc is 5 The average unit logistics time LT is 0.004 days/km.

Table 3. Test case information.

Case	Total Number of Tasks	Range of Number of Subtasks	Total Number of Subtasks	Number of Manufacturing Services
1	5	1–5	18	3
2	10	1–7	49	5
3	15	3–7	76	5
4	20	4–8	127	10
5	40	4–8	254	15
6	60	4–8	381	20

5.2. Algorithm Parameters and Performance Metrics

5.2.1. Setting Parameters

To obtain a higher-quality solution set with ISGA, it is necessary to determine the values of parameters, such as the population size, ζ, η, θ_1, and θ_2. Considering four factor levels and four parameters, a Taguchi orthogonal experiment was employed to determine the optimal parameter combinations. An orthogonal array L_{16} (4^4) was used, and the experiments were conducted using the medium-sized case #03 for comparison. In order to better judge the merits of the parameters, Wang et al. [29] proposed the average response variable (ARV). In each repeated experiment, multiple Pareto nondominated sets were obtained based on different parameter combinations, all Pareto solutions were combined, and Pareto solutions were selected from them to form a new Pareto nondominated set M*. The ratio of the number of solutions belonging to the new nondominated solution set M* under different parameter combinations is calculated; the larger the value, the better the parameter combination. In order to determine the effective initial population size, the ISGA is run independently 20 times on the basis of other parameters with constant values, for which the variation rate is 0.2, the maximum number of iterations is 500, and the difference in each population size is 50. And based on the suggestion of Han et al. [30], the clustering number K is set to 10 to run the ISGA independently 20 times. The ARVs for different parameter combinations are shown in Table 4.

According to the data in the table, the following changes in the ARV for different values of the four parameters are derived, and to better see the changes in the data, the data are represented in Figure 9.

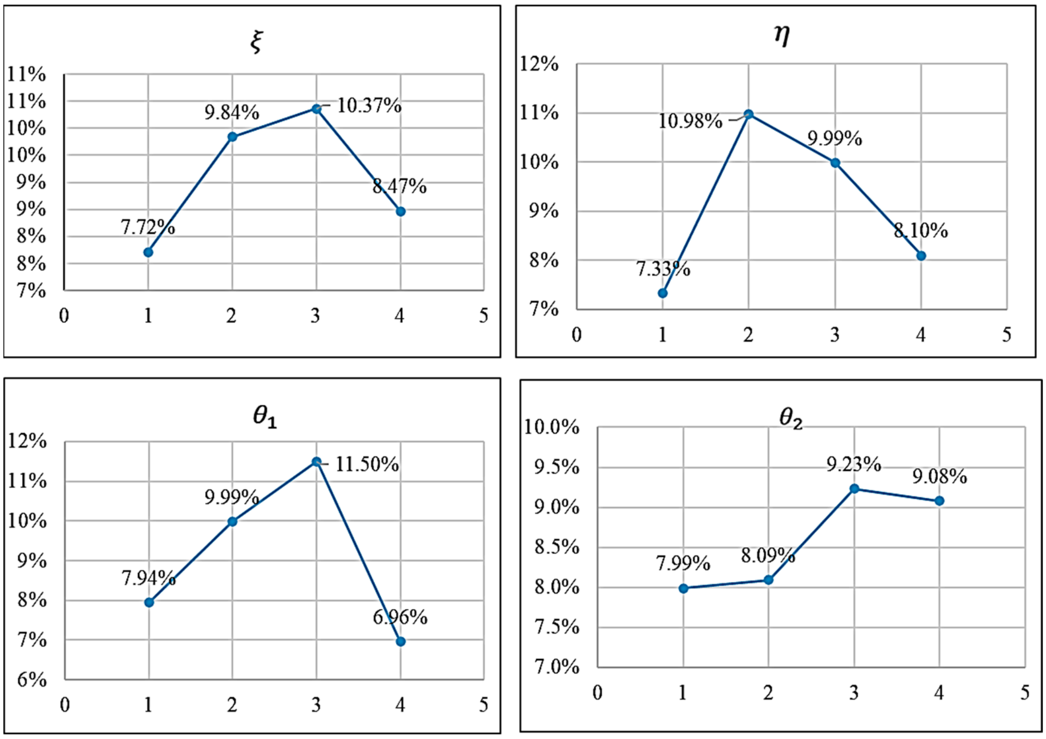

Figure 9. ARVs for different values.

Table 4. ARVs under orthogonal experiments.

Serial Number	ξ	η	θ_1	θ_2	ARV
1	1	1	1	1	5.06%
2	1	2	2	2	9.61%
3	1	3	3	3	12.64%
4	1	4	4	4	3.55%
5	2	1	2	3	8.09%
6	2	2	1	4	12.04%
7	2	3	4	1	9.61%
8	2	4	3	2	9.61%
9	3	1	3	4	11.12%
10	3	2	4	3	9.61%
11	3	3	1	2	8.09%
12	3	4	2	1	12.64%
13	4	1	4	2	5.06%
14	4	2	3	1	12.64%
15	4	3	2	4	9.61%
16	4	4	1	3	6.58%

5.2.2. Pareto Front

This paper establishes a multi-objective optimisation model of minimising workload and minimising energy consumption, due to these two optimisation objectives belong to the non-linear relationship, so this paper adopts the method of solving the Pareto-optimal solution set for its calculation, and the image of the Pareto-optimal solution set plotted on the axes of the objective function is the Pareto frontier. In order to better verify the effectiveness of the SGA algorithm, it is compared with the Hybrid Artificial Bee Colony Algorithm (TABC) [31], Modified Genetic Algorithm (MGA) [19], Pareto-based Grouping Discrete Harmony Search algorithm (PGDH) [32], and Multi-objective Particle Warm Optimisation algorithm (MOPSO) [33]

The results of the six random instances are shown in Table 5. From the data in the table, it can be seen that among the solution results of the five algorithms under different algorithms, ISGA takes the best value. Additionally, Figure 10 illustrates that the Pareto frontiers achieved by the ISGA algorithm are superior in comparison to those obtained by the other algorithms. That is, the Pareto solutions obtained by the other four algorithms are almost dominated by the solution set obtained by ISGA.

Table 5. Pareto-optimal solutions for different algorithms.

	Workload					Satisfaction				
	ISGA	MGA	MOPSO	ABC	HS	ISGA	MGA	MOPSO	ABC	HS
1	**28**	43.2	46.8	41.6	56.8	**0.636**	0.639	0.647	0.653	0.724
2	**8.8**	12	7.6	11.2	26.4	**0.555**	0.577	0.574	0.578	0.701
3	**5.6**	32.4	20	10	60	**0.526**	0.592	0.562	0.582	0.632
4	**21.2**	91.6	140	41.2	240	**0.588**	0.642	0.67	0.592	0.700
5	**103**	256.1	230	137	428	**0.621**	0.719	0.713	0.671	0.765
6	**545.3**	703.9	549.1	628.6	841	**0.724**	0.868	0.8473	0.7631	0.901

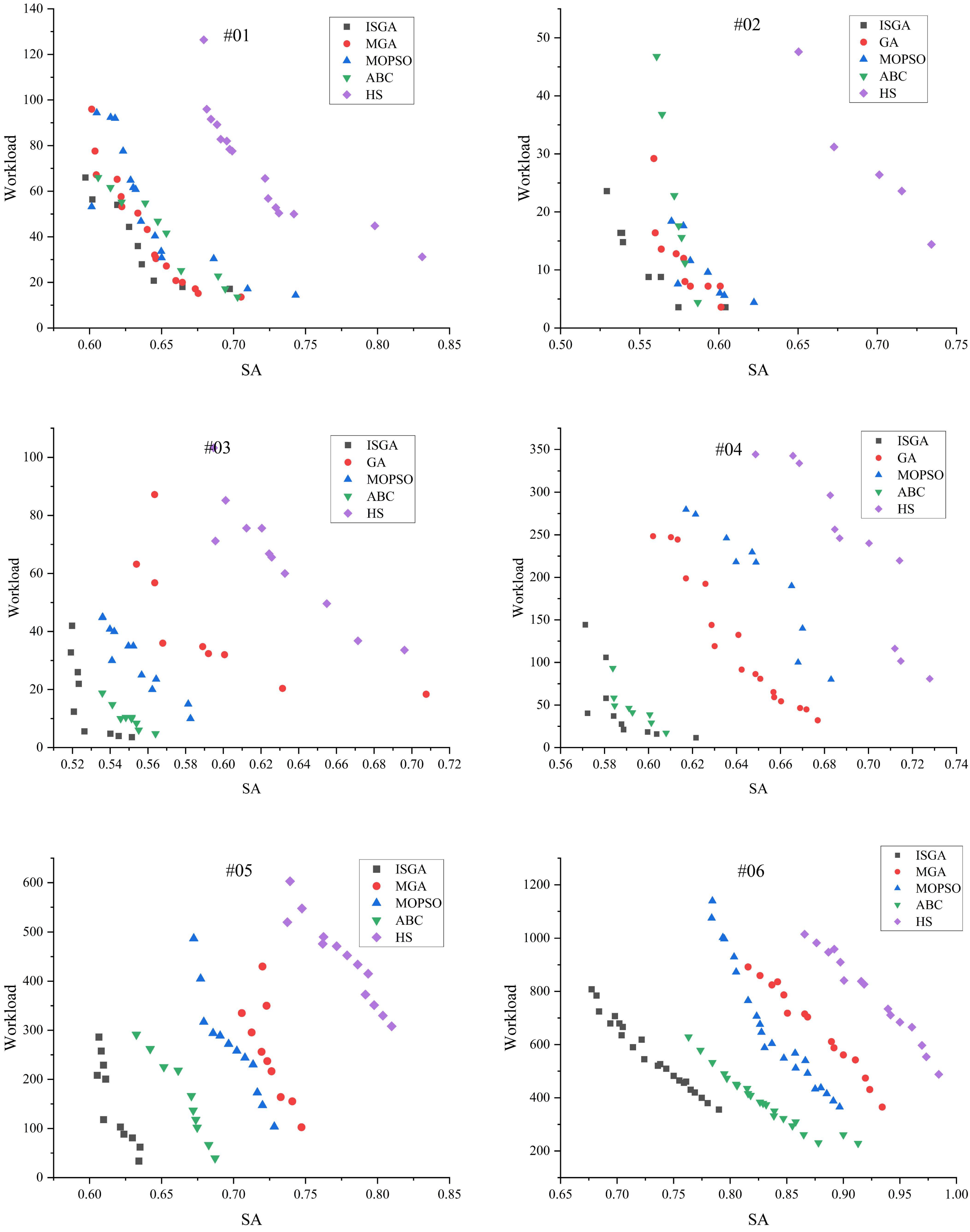

Figure 10. Pareto frontiers for different algorithms under different arithmetic cases.

5.2.3. Evaluation Indicators

In order to verify the effectiveness of the improved algorithm proposed in this paper, the distance with the Lower Bound (*DLB*), the proximity of the Pareto bound to the Lower Bound, for which the smaller value represents the better solution set, is introduced, adapted from Mansouri et al. [34], and the calculation method is shown in Equations (24) and (25):

$$DLB_W = \frac{\sum\left(W_i - W_{min}/W_{min}\right)}{\Omega} \tag{24}$$

$$DLB_{SA} = \frac{\sum\left(SA_i - SA_{max}/SA_{max}\right)}{\Omega} \tag{25}$$

Here, Ω denotes the set of all Pareto solutions obtained by the algorithms after 20 runs, and W_{min} and SA_{max} denote the minimum workload and maximum satisfaction, i.e., the lower bounds of the model solutions obtained by the five algorithms, respectively. W_i and SA_i denote the ith solution in the algorithms' Pareto frontiers, where i belongs to Ω. For different problem sizes, the smaller the value of the *DLB*, the better the set of Pareto solutions obtained by the ISGA. Under different problem sizes, the smaller the value of *DLB*, the better the set of Pareto solutions it obtains, so it is obvious that ISGA always outperforms the other four algorithms in solving the model, as shown in Table 6.

Table 6. DLB of different algorithms in different cases.

	Workload					**Satisfaction**				
	ISGA	**MGA**	**MOPSO**	**ABC**	**HS**	**ISGA**	**MGA**	**MOPSO**	**ABC**	**HS**
1	**1.784**	64.07	89.49	66.8	119.101	**0.064**	0.079	0.08	0.094	0.205
2	3.059	**2.256**	7.806	5.159	6.956	**0.049**	0.09	0.11	0.083	0.313
3	**3.728**	10.765	7.431	11.889	17.263	**0.021**	0.149	0.067	0.059	0.213
4	**3.144**	7.298	16.024	6.026	19.207	**0.031**	0.134	0.137	0.039	0.21
5	**1.625**	3.084	3.769	2.0522	6.3729	**0.017**	0.198	0.146	0.090	0.267
6	**0.9910**	1.8905	2.0524	0.8118	2.9320	**0.0931**	0.2913	0.2298	0.2090	0.2830

Due to the non-compliance of the data distribution with the assumptions of a standard *t*-test, a Robust *t*-test, also known as the Wilcoxon Signed-Rank Test, was employed. It assesses the differences between our proposed algorithm and the other four algorithms in terms of experimental results. If a *p*-value is less than 0.05, the differences are considered significant. Table 7 presents the results of the Robust *t*-test, displaying the corresponding *p*-values. For most instances, the *p*-values are below 0.05, indicating significant differences between the ISGA and the other four algorithms. Thus, the effectiveness of the ISGA is further confirmed.

Table 7. Robust *t*-test results.

	#01	**#02**	**#03**	**#04**	**#05**	**#06**
MGA	1.249×10^{-5}	0.0028089	0.01	2.558×10^{-5}	0.0199	2.58×10^{-5}
MOPSO	0.00122	0.001289	0.002	0.01198	0.0199	3.25×10^{-5}
ABC	0.0061	0.01551	0.010	5.01×10^{-6}	0.0013	0.0009
HS	0.0146	0.01469	0.0250	1.104×10^{-5}	3.14×10^{-8}	1.91×10^{-8}

To further assess the efficacy of the ISGA algorithm, this section additionally employs the Inverted Generational Distance (IGD) metric. This metric considers not only the proximity of the generated solutions to the true Pareto front but also the uniformity in the distribution of these solutions. According to the data presented in Table 8, the ISGA significantly outperforms other algorithms in terms of IGD metrics.

Table 8. IGD values under different algorithms.

IGD	#01	#02	#03	#04	#05	#06
ISGA	0.0227	**0.0124**	**0.8860**	**3.4819**	**0.0059**	**0.01079**
MGA	2.6211	0.0132	5.1583	9.9330	8.3972	6.90664
MOPSO	5.9394	0.0444	2.7335	25.1119	23.5035	0.68788
ABC	**0.0217**	0.0438	1.2324	5.2528	0.5877	0.0230
HS	10.8718	0.7701	9.5462	25.3649	76.5652	28.71873

5.3. Results Analysis and Discussion

In this subsection, we examine the Pareto solution for the ISGA in ten order tasks. Detailed manufacturing resource information is shown in Tables 9 and 10. The processing and manufacturing sequences corresponding to partial orders, and the required manufacturing services are shown in Figure 11.

Table 9. Manufacturer and manufacturing resource information.

Manufacturers	Service	Time/min	Processing Cost/$	Set Cost/$	Quality (%)
1	turn	12	29	15	92
	drilling	33	17	24	86
	soldered	28	18	19	95
	planning	19	20	16	94
2	bores	23	23	20	82
	soldered	17	18	22	95
	turn	35	15	19	92
3	punch	18	18	17	89
	mill	26	17	23	95
	soldered	34	17	21	94
4	planning	12	27	16	92
	turn	26	11	22	83
	drilling	38	20	25	94
5	bores	36	23	24	95
	planning	16	18	23	98
	soldered	20	24	19	92

Table 10. Distance between manufacturers.

km	1	2	3	4	5	6	7	8	9	10
1	0	355.4	245.5	272.3	20.2	126.2	55.2	10.8	19.3	170.3
2	355.4	0	100.2	21.5	153.9	200.2	24.5	15.4	148.9	298.1
3	245.5	100.2	0	292.6	60.2	28.7	73.2	22.2	21.1	18.4
4	272.3	21.5	292.6	0	113.5	175.8	178.1	301.6	156.9	279.3
5	20.2	153.9	60.2	113.5	0	41.6	16.5	121	18.6	16
6	126.2	200.2	28.7	175.8	41.6	0	19.6	63.1	92.4	186.2
7	55.2	24.5	73.2	178.1	16.5	19.6	0	16.6	48.9	32.5
8	10.8	15.4	22.2	301.6	121	63.1	16.6	0	102.7	64.3
9	19.3	148.9	21.1	156.9	18.6	92.4	48.9	102.7	0	136.1
10	170.3	298.1	18.4	279.3	16	186.2	32.5	64.3	136.1	0

Different automotive parts require different processing services for combined manufacturing. According to the execution process of the cloud platform, after receiving the order information, the cloud platform will pre-process the order, decompose each manufacturing task into multiple sub-tasks with different granularities, and determine its candidate service set. Table 11 shows the processing information after decomposition.

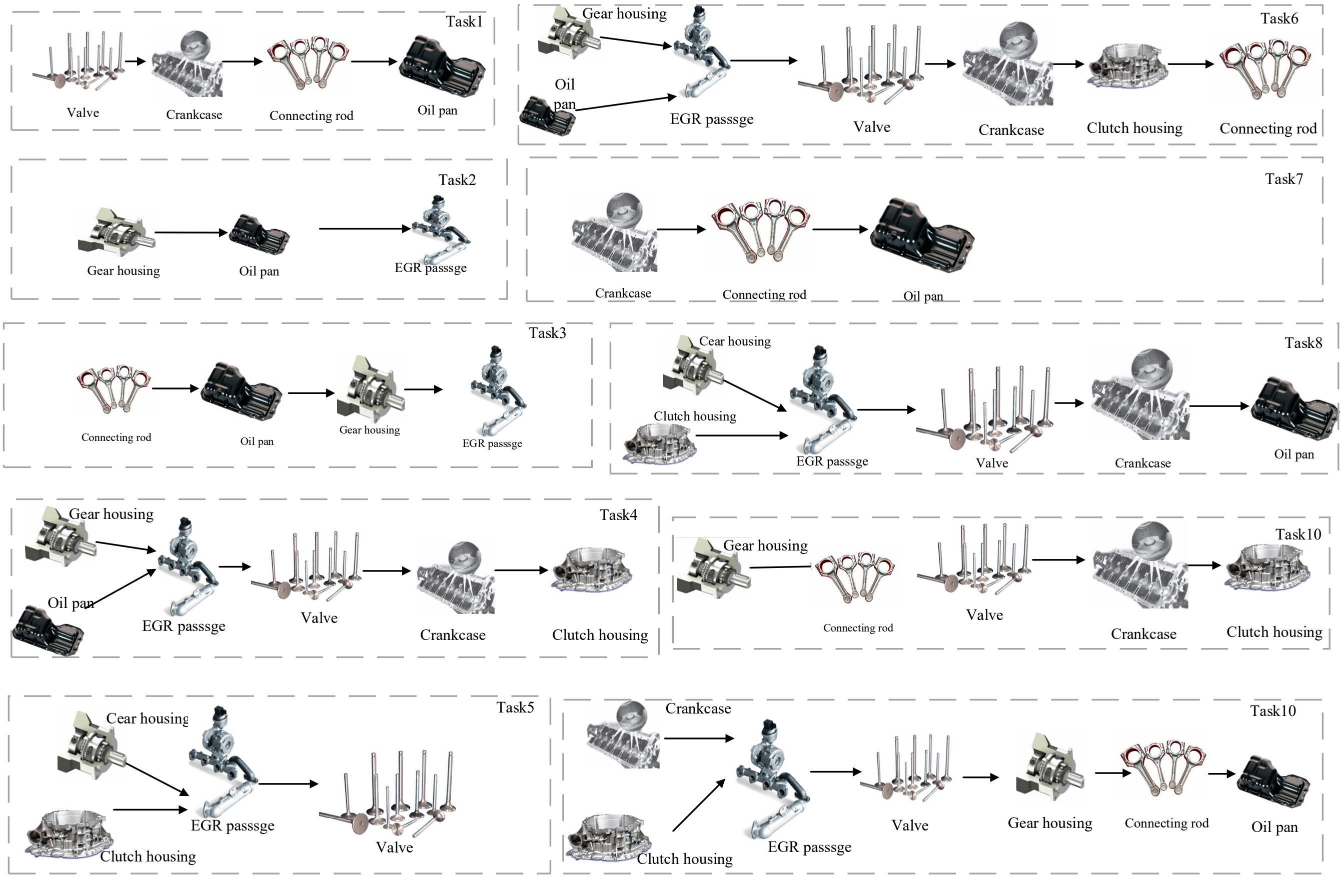

Figure 11. Order processing sequence.

Table 11. Scheduling task information.

Task	Number of Subtasks	Subtask
1	4	Valve/Crankcase/Connecting rod/Oil pan
2	3	Gear housing/Oil pan/EGR passage
3	4	Connecting rod/Oil pan/Gear housing/EGR passage
4	6	Gear housing/Oil pan/EGR passage/Valve/Crankcase/Clutch housing
5	4	Gear housing/Clutch housing/EGR passage/Valve
6	7	Gear housing/Oil pan/Valve/Crankcase/Clutch housing/Connecting rod
7	3	Crankcase/Connecting rod/Oil pan
8	6	Gear housing/Clutch housing/EGR passage/Valve/Crankcase/Oil pan
9	5	Gear housing/Connecting rod/Valve/Crankcase/Clutch housing
10	7	Crankcase/Clutch housing/EGR passage/Valve/Gear housing/Oil pan

Figure 12 shows the Pareto frontier for cloud manufacturing scheduling optimisation for both the supply and demand sides.

The solution closest to the origin in the Pareto frontier is taken, and its corresponding subtask processing order and the corresponding processing and manufacturing service scheduling scheme are shown in Figure 13. Based on the Pareto solution set, it can be concluded that the range of workloads is between 7.2 and 99.2 min and customer satisfaction is between 0.59 and 0.66. From the manufacturing service provider's point of view, at the expense of customer satisfaction, the manufacturer can achieve a minimum workload of 7.2 min, which is a 92% reduction in the minimum workload and a significant improvement in resource utilisation. At the same time, every solution in the set of Pareto solutions is satisfying the customer. That is to say, the manufacturing service provider can choose the scheduling solution according to the amount of free resources under the condition of satisfying the demand side of manufacturing services.

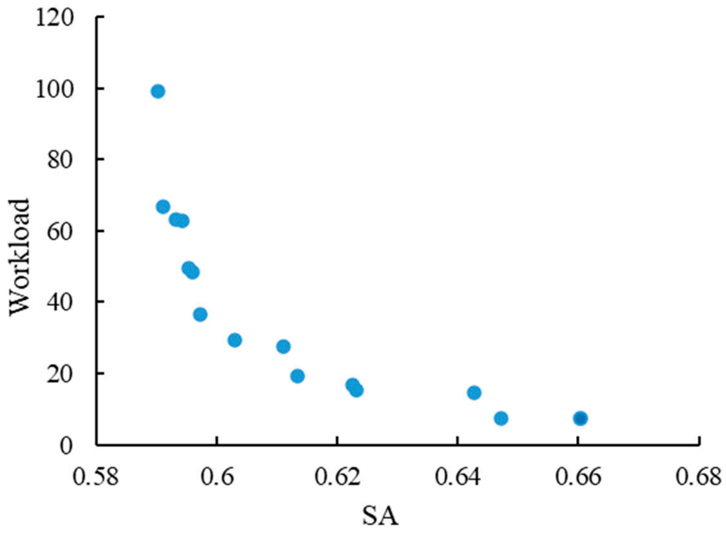

Figure 12. Case Pareto frontier for supply and demand sides.

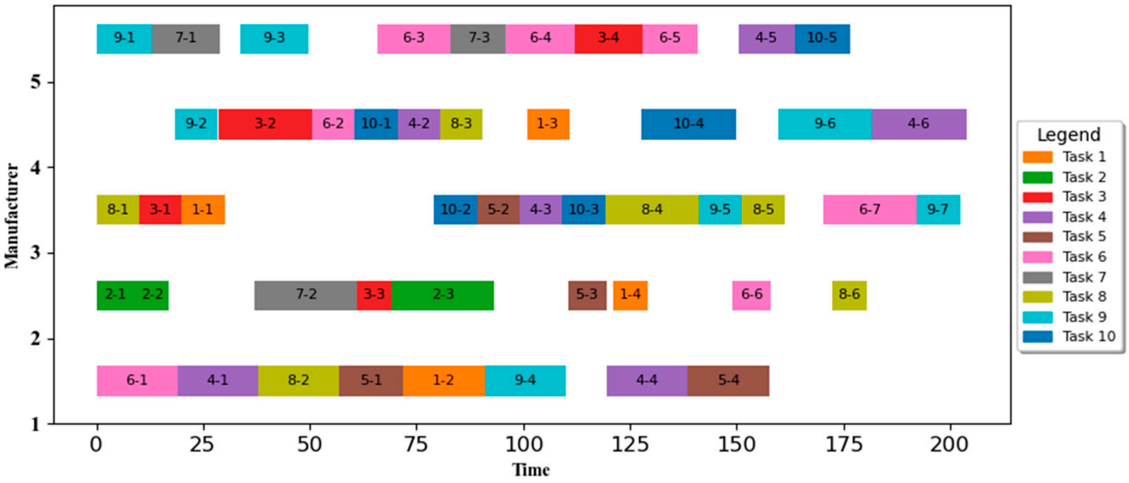

Figure 13. Supply and demand bilateral scheduling optimisation results Gantt chart.

Similarly, to better illustrate the necessity of scheduling optimisation research for both supply and demand sides, the ISGA is used for solving the problem, and the optimisation objectives are set to o total completion time, total cost, and total quality, while using the case data. The results of 20 scheduling optimisation solutions are summarised, from which the optimal solution is taken out and a Gantt chart is drawn based on the obtained scheduling. As shown in Figure 14, the blank space between processes indicates that the manufacturer's processing machine is in an idle state, and comparing the two Gantt charts, it can be seen that the workload is more balanced under the resource scheduling optimisation model oriented towards the supply and demand sides; the resource utilisation rate under the two scenarios is shown in Table 12, and when load balancing is taken into account, the lowest resource utilisation rate of the manufacturing service provider is 71.20%, the highest reaches 93.94%, and the average resource utilisation rate is 76%, while the highest resource utilisation rate of the manufacturer is only 72.56% when load balancing is not considered, and the resource utilisation rate of some manufacturers is less than half, with an average resource utilisation rate of 62.83%. This shows that resource utilisation can be improved when load balancing is considered.

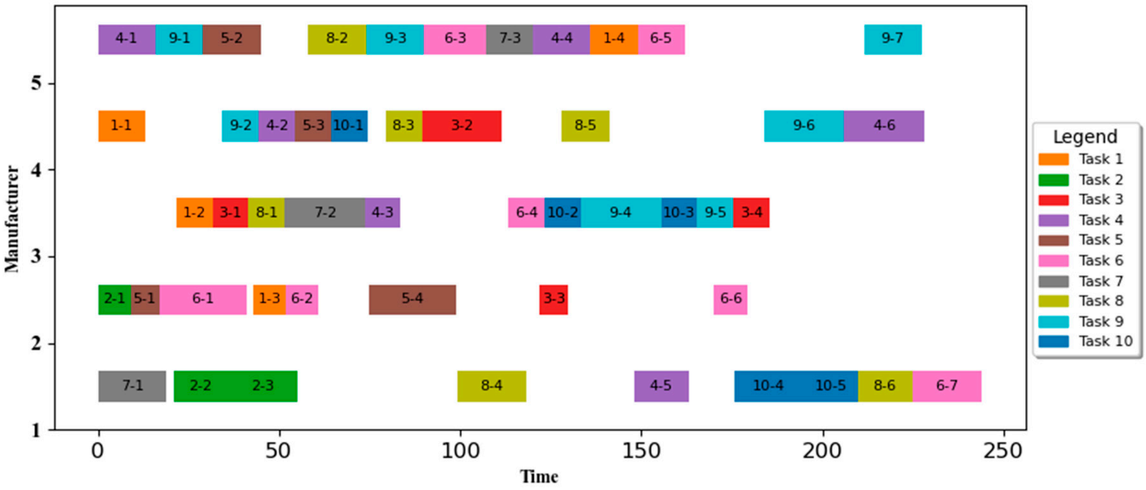

Figure 14. Gantt chart of optimisation results without considering load balancing scheduling.

Table 12. Whether to consider load balancing resource utilisation comparison.

	Consider Load Balancing					Do Not Consider Load Balancing				
Task	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
Load Time	148	107	144	148	146	155	100	134	142	165
Total Boot Time	157.5	180.4	202.2	203	176.6	234.7	243.7	185.4	228	227.4
Resource Utilisation	93.9%	59.3%	71.2%	72.9%	82.6%	66.%	41.%	72.2%	62.3%	72.5%
Average Utilisation			76.00%					62.83%		

6. Conclusions and Outlook

The research objective of this paper is to solve the optimisation of resource scheduling to satisfy both the supply and demand sides of the service in a cloud manufacturing environment and then to achieve better resource utilisation. Firstly, the demand and supply sides are considered, and the optimisation objectives of minimising load balancing and maximising customer satisfaction are taken as the optimisation objectives, so that the scheduling scheme can meet the demands of both supply and demand sides. Secondly, an improved genetic algorithm is designed, which adopts a staged solution method, while the k-means algorithm is integrated into the selection operator to improve the diversity of the population; a real-time sequential scheduling strategy is adopted, which improves the quality of the understanding. Finally, by comparing and analysing with the other four heuristic algorithms, the effectiveness and efficiency of the ISGA in solving the supply-demand dual-issue coordinated scheduling problem is verified.

This paper lays theoretical foundations and provides empirical insights for multi-objective multi-task scheduling optimisation in cloud manufacturing environments. Nevertheless, given the diversity of cloud manufacturing scenarios, substantial research opportunities remain. Future efforts should encompass a broader array of scheduling objectives, including carbon emissions, water usage, labour costs, and order batching, to address the cloud manufacturing multi-task scheduling challenge more holistically. Additionally, incorporating dynamic factors, such as machine breakdowns, rush orders, and schedule changes, into the scheduling process is crucial, enabling rapid task reassignment and flexible priority adjustments to accommodate evolving production demands.

Author Contributions: Conceptualisation, Q.Z.; Methodology, J.D.; Resources, Y.Z.; Writing—original draft, N.L.; Writing—review and editing, J.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data are contained within the article, and the corresponding author can be contacted if there are other requests.

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

Appendix A. Pseudo-Code for Adaptive Population Size Algorithm

Algorithm A1: Pareto-based adaptive population size method

```
For $i$ = 1: $SN$
        If $X_i$ dominated by $X_i'$
                Reserve $X_i'$ into the next generation
        If $X_i'$ is dominated by $X_i$
                Reserve $X_i$ into the next generation
Else
    Calculate success rate SR of all individuals
    If SR > $rand$ // shrink the population
        Calculate success rate $sr_i$ of the ith individual
            If $X_i \epsilon G$ && $sr_i$ < $rand$ && $SN$> $SN_{\min}$
                $SN = SN - 1$
            End if
    Else // expand the population
            If $SN < SN_{\max}$
                    Reserve $X_i'$ into the next generation and $SN = SN + 1$
            End if
End for
        Update $M$ and $G$
```

Appendix B. Clustering Algorithm Kmeans Pseudo-Code

Algorithm A2: Procedure of k-means cluster in selection operator

```
Begin
    Normalised the objective function of each individual $p$
    Choose Kindividuals randomly as the centre of clusters, set $iter = 1,\ E_0 = 0$
    While $iter \leq C_{\text{maxiter}}$
            Calculate the Euclidean distance between the individuals and $K$ centres
            Assign each individual to a cluster based on the smallest Euclidean distance.

        Calculate the sum of the squared errors $E_{iter} = \sum_{p=1}^{sop} \sum_{k=1}^{K} z_{pk} \left| \overline{f_p} - \mu_k \right|^2,$
        Where $\mu_k$ denotes objective function vector of the $k$th cluster centre and sop =
size of population. $z_{pk} = 1,$ if individual $p$ in the cluster $C_k,$ else $z_{pk} = 0.$

    If $iter > 1$
            If $E_{iter} = E_{iter-1}$
                break.
            Else
                Update its centre $\mu_k$.
            Endif
    Endif
    $iter = ter + 116.$
    Endwhile
End
```

Appendix C. Real-Time Task Sequence Scheduling Pseudo-Code

Algorithm A3: Real-Time sequence scheduling

```
While $h \leq \sum_{u=1}^{U} K_u$ // $h$ is the total number of sequenced subtasks
    If the subtasks are newly available at the current time
        Select the newly available subtasks
        Calculate the laxity of the selected subtasks
        Sequence the selected subtasks according to the laxity value
        If two or more selected subtasks have the same laxity value
            Sequence the selected subtasks that have the same laxity value randomly
        End if
        Add the sequence record to the subtask sequence vector
      $h = h + \delta$ // $\delta$ is the number of selected subtasks at each time
End while
```

References

1. Rashidifar, R.; Bouzary, H.; Chen, F.F. Resource Scheduling in Cloud-Based Manufacturing System: A Comprehensive Survey. *Int. J. Adv. Manuf. Technol.* **2022**, *122*, 4201–4219. [CrossRef]
2. Bouzary, H.; Chen, F.F.; Krishnaiyer, K. Service Matching and Selection in Cloud Manufacturing: A State-of-the-Art Review. *Procedia Manuf.* **2018**, *26*, 1128–1136. [CrossRef]
3. Liu, B.; Zhang, Z. QoS-Aware Service Composition for Cloud Manufacturing Based on the Optimal Construction of Synergistic Elementary Service Groups. *Int. J. Adv. Manuf. Technol.* **2017**, *88*, 2757–2771. [CrossRef]
4. Goodarzi, E.V.; Houshmand, M.; Valilai, O.F.; Ghezavati, V.; Bamdad, S. Manufacturing Cloud Service Composition Based on the Non-Cooperative and Cooperative Game Theory. In Proceedings of the 2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, 14–17 December 2020; pp. 1122–1125.
5. Qian, C.; Zhang, Y.; Liu, Y.; Wang, Z. A Cloud Service Platform Integrating Additive and Subtractive Manufacturing with High Resource Efficiency. *J. Clean. Prod.* **2019**, *241*, 118379. [CrossRef]
6. Valizadeh, S.; Fatahi Valilai, O.; Houshmand, M. Flexible Flow Line Scheduling Considering Machine Eligibility in a Digital Dental Laboratory. *Int. J. Prod. Res.* **2020**, *58*, 6513–6531. [CrossRef]
7. Jian, C.F.; Wang, Y. Batch Task Scheduling-Oriented Optimization Modelling and Simulation in Cloud Manufacturing. *Int. J. Simul. Model.* **2014**, *13*, 93–101. [CrossRef]
8. Yang, D.; Liu, Q.; Li, J.; Jia, Y. Multi-Objective Optimization of Service Selection and Scheduling in Cloud Manufacturing Considering Environmental Sustainability. *Sustainability* **2020**, *12*, 7733. [CrossRef]
9. Zeynivand, M.; Ranjbar, H.; Radmanesh, S.-A.; Fatahi Valilai, O. Alternative Process Routing and Consolidated Production-Distribution Planning with a Destination Oriented Strategy in Cloud Manufacturing. *Int. J. Comput. Integr. Manuf.* **2021**, *34*, 1162–1176. [CrossRef]
10. Hu, Y.; Zhu, F.; Zhang, L.; Lui, Y.; Wang, Z. Scheduling of Manufacturers Based on Chaos Optimization Algorithm in Cloud Manufacturing. *Robot. Comput.-Integr. Manuf.* **2019**, *58*, 13–20. [CrossRef]
11. Li, W.; Zhu, C.; Yang, L.T.; Shu, L.; Ngai, E.C.-H.; Ma, Y. Subtask Scheduling for Distributed Robots in Cloud Manufacturing. *IEEE Syst. J.* **2017**, *11*, 941–950. [CrossRef]
12. Tong, H.; Zhu, J. A Novel Method for Customer-Oriented Scheduling with Available Manufacturing Time Windows in Cloud Manufacturing. *Robot. Comput.-Integr. Manuf.* **2022**, *75*, 102303. [CrossRef]
13. 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]
14. Zhou, J.; Yao, X. A Hybrid Approach Combining Modified Artificial Bee Colony and Cuckoo Search Algorithms for Multi-Objective Cloud Manufacturing Service Composition. *Int. J. Prod. Res.* **2017**, *55*, 4765–4784. [CrossRef]
15. Delaram, J.; Valilai, O.F. A Mathematical Model for Task Scheduling in Cloud Manufacturing Systems Focusing on Global Logistics—ScienceDirect. *Procedia Manuf.* **2018**, *17*, 387–394. [CrossRef]
16. Zhang, W.; Ding, J.; Wang, Y.; Zhang, S.; Xiong, Z. Multi-Perspective Collaborative Scheduling Using Extended Genetic Algorithm with Interval-Valued Intuitionistic Fuzzy Entropy Weight Method. *J. Manuf. Syst.* **2019**, *53*, 249–260. [CrossRef]
17. Zheng, H.; Feng, Y.; Tan, J. A Hybrid Energy-Aware Resource Allocation Approach in Cloud Manufacturing Environment. *IEEE Access* **2017**, *5*, 12648–12656. [CrossRef]
18. Zhou, L.; Zhang, L.; Fang, Y. Logistics Service Scheduling with Manufacturing Provider Selection in Cloud Manufacturing. *Robot. Comput.-Integr. Manuf.* **2020**, *65*, 101914. [CrossRef]
19. Li, F.; Zhang, L.; Liao, T.W.; Liu, Y. Multi-Objective Optimisation of Multi-Task Scheduling in Cloud Manufacturing. *Int. J. Prod. Res.* **2019**, *57*, 3847–3863. [CrossRef]
20. Zhou, L.; Zhang, L.; Horn, B.K.P. Collaborative Optimization for Logistics and Processing Services in Cloud Manufacturing. *Robot. Comput.-Integr. Manuf.* **2021**, *68*, 102094. [CrossRef]

21. Wu, Q.; Xie, N.; Zheng, S. Integrated Cross-Supplier Order and Logistic Scheduling in Cloud Manufacturing. *Int. J. Prod. Res.* **2022**, *60*, 1633–1649. [CrossRef]
22. Helo, P.; Phuong, D.; Hao, Y. Cloud Manufacturing—Scheduling as a Service for Sheet Metal Manufacturing. *Comput. Oper. Res.* **2019**, *110*, 208–219. [CrossRef]
23. Ahn, G.; Hur, S. Multiobjective Real-Time Scheduling of Tasks in Cloud Manufacturing with Genetic Algorithm. *Math. Probl. Eng.* **2021**, *2021*, 1305849. [CrossRef]
24. Jafarnejad Ghomi, E.; Masoud Rahmani, A.; Nasih Qader, N. Service Load Balancing, Task Scheduling and Transportation Optimisation in Cloud Manufacturing by Applying Queuing System. *Enterp. Inf. Syst.* **2019**, *13*, 865–894. [CrossRef]
25. Li, F.; Liao, T.W.; Zhang, L. Two-Level Multi-Task Scheduling in a Cloud Manufacturing Environment. *Robot. Comput.-Integr. Manuf.* **2019**, *56*, 127–139. [CrossRef]
26. Yuan, M.; Cai, X.; Zhou, Z.; Sun, C.; Gu, W.; Huang, J. Dynamic Service Resources Scheduling Method in Cloud Manufacturing Environment. *Int. J. Prod. Res.* **2021**, *59*, 542–559. [CrossRef]
27. Shi, Y.; Luo, L.; Guang, H. Research on Scheduling of Cloud Manufacturing Resources Based on Bat Algorithm and Cellular Automata. In Proceedings of the 2019 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE), Hangzhou, China, 20–21 April 2019; pp. 174–177.
28. Cui, L.; Li, G.; Zhu, Z.; Lin, Q.; Wen, Z.; Lu, N.; Wong, K.-C.; Chen, J. A Novel Artificial Bee Colony Algorithm with an Adaptive Population Size for Numerical Function Optimization. *Inf. Sci.* **2017**, *414*, 53–67. [CrossRef]
29. Wang, L.; Wang, S.; Liu, M. A Pareto-Based Estimation of Distribution Algorithm for the Multi-Objective Flexible Job-Shop Scheduling Problem. *Int. J. Prod. Res.* **2013**, *51*, 3574–3592. [CrossRef]
30. Han, J.; Kamber, M.; Pei, J. Methods. In *Data Mining: Concepts and Techniques*; Elsevier: Amsterdam, The Netherlands, 2012; pp. 443–495. ISBN 978-0-12-381479-1.
31. Li, J.; Pan, Q. Solving the Large-Scale Hybrid Flow Shop Scheduling Problem with Limited Buffers by a Hybrid Artificial Bee Colony Algorithm. *Inf. Sci.* **2015**, *316*, 487–502. [CrossRef]
32. Gao, K.Z.; Suganthan, P.N.; Pan, Q.K.; Chua, T.J.; Cai, T.X.; Chong, C.S. Pareto-Based Grouping Discrete Harmony Search Algorithm for Multi-Objective Flexible Job Shop Scheduling. *Inf. Sci.* **2014**, *289*, 76–90. [CrossRef]
33. Abdullah, M.; Al-Muta'a, E.A.; Al-Sanabani, M. Integrated MOPSO Algorithms for Task Scheduling in Cloud Computing. *J. Intell. Fuzzy Syst.* **2019**, *36*, 1823–1836. [CrossRef]
34. Mansouri, S.A.; Aktas, E.; Besikci, U. Green Scheduling of a Two-Machine Flowshop: Trade-off between Makespan and Energy Consumption. *Eur. J. Oper. Res.* **2016**, *248*, 772–788. [CrossRef]

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