Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 20 (2): 3120–3145.
DOI: 10.3934/mbe.2023147
Received: 16 September 2022
Revised: 31 October 2022
Accepted: 16 November 2022
Published: 02 December 2022

Research article

Modeling and simulation of task rescheduling strategy with resource substitution in cloud manufacturing

Xiaodong Zhang and Dawei Ren*

School of Economics and Management, University of Science and Technology Beijing, Beijing 100083, China

* **Correspondence:** Email: dw_ren@126.com.

Abstract: When a cloud manufacturing environment extends to multi-user agent, multi-service agent and multi-regional spaces, the process of manufacturing services faces increased disturbances. When a task exception occurs because of disturbance, it is necessary to quickly reschedule the service task. We propose a multi-agent simulation modeling approach to simulate and evaluate the service process and task rescheduling strategy of cloud manufacturing, with which impact parameters can be achieved through careful study under different system disturbances. First, the simulation evaluation index is designed. In addition to the quality of service index of cloud manufacturing, the adaptive ability of task rescheduling strategy in response to a system disturbance is considered, and the flexibility of cloud manufacturing service index is proposed. Second, considering the substitution of resources, the internal and external transfer strategies of service providers are proposed. Finally, a simulation model of the cloud manufacturing service process of a complex electronic product is constructed by multi-agent simulation, and simulation experiments under multiple dynamic environments are designed to evaluate different task rescheduling strategies. The experimental results indicate that the external transfer strategy of the service provider in this case has higher quality of service and flexibility of service. Sensitivity analysis indicates that the matching rate of substitute resources for internal transfer strategy of service providers and the logistics distance of external transfer strategy of service providers are both sensitive parameters, which have significant impacts on the evaluation indexes.

Keywords: cloud manufacturing; quality of service; flexibility of service; task rescheduling; multi-agent simulation

1. Introduction

With the development of Internet, information and manufacturing technologies, the manufacturing model began to change from a large-scale production mode to a user-oriented service mode. Cloud manufacturing (CMfg) is a new service-oriented manufacturing mode proposed in this background [1]. Li et al. [2] defined CMfg as a new networked manufacturing mode, where various online manufacturing resources are organized in an orderly manner on the cloud platform, and users can access the network to obtain qualified and satisfactory manufacturing services. Since the manufacturing environment expands to multi-user agent, multi-service agent and multi-regional spaces, CMfg inevitably faces higher uncertainties, such as more frequent changes in user requirements [3], manufacturing resource failures and increased susceptibility to interference in logistics and transportation [4]. Task exception is one of the specific manifestations of uncertainty in CMfg. When the CMfg service platform experiences disturbances such as an emergency insertion order, a manufacturing service resource failure, poor logistics transportation, etc., the established manufacturing task cannot be completed in the expected time, and a series of chain reactions is triggered. Therefore, task exceptions can occur where the manufacturing network is weak. When a task exception occurs, it is necessary to quickly reschedule the production system. At present, the approaches of CMfg system rescheduling include dynamic scheduling of service composition and system simulation evaluation.

Dynamic scheduling of service combinations is a resolution approach that is currently widely used; it consists of the construction of a rescheduling model based on the initial static scheduling model. The new service composition scheme is taken as the decision variable; service time, cost, reliability or comprehensive quality of service is taken as the optimization objective [5]. The constraints such as order completion time and resource occupation are also considered to establish a mathematical model, which is solved by using various optimization algorithms [6]. Although dynamic scheduling ensures that the rescheduling scheme is still at an optimized level through mathematical programming, the newly generated scheduling scheme is a global adjustment to subsequent tasks, which is larger than the adjustment for the original production plan and affects more users and service providers. In distributed cloud services, tasks caused by dynamic perturbations occur abnormally, and the frequent dynamic adjustments brought about by them make it difficult for service providers to operate realistically, owing to the following reasons. (1) After the initial CMfg plan is issued, the service provider needs to prepare the received tasks in advance to ensure the smoothness of the entire manufacturing chain. Frequent dynamic task adjustments can lead to the failure of existing preparations and insufficient preparations for new tasks, which not only create additional task processing time but also involve a series of issues such as manufacturing costs and procurement changes. (2) Although the adjusted production plan is still at an optimized level for the current cloud platform production, the distributed CMfg service providers must complete certain internal production tasks in addition to undertaking CMfg service tasks. This situation restricts the practical application of dynamic scheduling of service composition in the CMfg service platform.

In the real CMfg platform, the strategy of local adjustment is suitably applied to solve production exceptions caused by system disturbances. Of course, this local adjustment strategy can also lead to overall changes in order execution and resource usage due to the dependencies between tasks. This implies the necessity of a simulation evaluation. Simulation can truly reflect the uncertainty of the manufacturing environment and the dynamic process of task rescheduling. Based on the scheme with the least impact on the actual production process, a rapid and dynamic response to production exceptions can be achieved through the operation and evaluation of various scheduling schemes.

Different from dynamic scheduling, system simulation evaluation is a dynamic adjustment approach based on experiments [7]. Based on the simulation model of the production system, this approach compares and evaluates the possible production recovery strategies with multiple schemes to find a solution to the production exception [8]. In simulation studies, the rescheduling strategy for production exceptions is usually assumed to manage the exception tasks with substitution resources, which requires the resources of the manufacturing system to have a certain degree of interchangeability. When task exceptions occur, the manufacturing system only transfers the exception tasks that have accumulated while the plans of other tasks remain unchanged as much as possible to avoid frequent global task reassignment.

So far, there are several studies on the simulation of manufacturing systems for exception tasks, but there are still few for the CMfg platform. Since multi-agent simulation is suitable for describing uncertainty [9], distribution and dynamics of the CMfg system [10], this paper proposes a multi-agent simulation modeling approach to simulate the service process and task rescheduling strategies of CMfg. The goal is to analyze the impacts of different task rescheduling strategies on the manufacturing system under various system disturbances. However, the task rescheduling strategy includes internal and external transfer strategies of service providers. In the simulation evaluation of the task rescheduling strategy, in addition to focusing on the typical performance indicators of the CMfg service, this paper also considers the adaptability of the task rescheduling strategy to system disturbance and proposes the flexibility of service index of CMfg. To evaluate the different task rescheduling strategies in a comprehensive way, simulation experiments in multiple uncertain environments are constructed on this basis.

The rest of this paper is structured as follows. Section 2 presents a systematic review of literature relevant to current research, including models, approaches and factors related to CMfg task rescheduling. Section 3 gives the evaluation indexes of the rescheduling strategy. Section 4 presents the construction approach and achievement process of the CMfg service multi-agent model. Section 5 designs comparative simulation experiments of rescheduling strategies under different disturbance degrees. Section 6 discusses simulation results and parameter sensitivity analysis. Finally, Section 7 presents the conclusions of this paper and perspectives of future research.

2. Related work

2.1. Dynamic scheduling of cloud manufacturing service composition

In CMfg, multiple services from different providers need to be composed to satisfy complex and diverse user orders. The scheduling problem of cloud manufacturing service composition (CMfg-SC) is to find the optimal service composition scheme for a given order. Whereas static scheduling is the optimization of the initial service composition scheme for the order, dynamic scheduling (also known as rescheduling) is to dynamically adjust the service composition scheme during the execution of the service. Most previous studies focused on static scheduling of cloud manufacturing service composition. Various models and algorithms have been proposed [11], including the scheduling programming model [12], multi-objective mathematical model [13] and approach [14], association analysis approach [15] and Ant Colony Optimization (ACO) algorithm [16]. There are currently only a few studies on dynamic scheduling of CMfg-SC [17]. Haleh et al. [18] proposed a control algorithm that utilizes dynamic task filtering based on the evaluation of task utilization to keep the service system running in a stable area. Liu et al. [19] proposed a real-time task scheduling approach for multi-agents based on the characteristics of cloud service scheduling and logistics. This approach is based on the unified management of SMA and the rescheduling of tasks, which can eliminate the impact of service exceptions in a timely manner. To reduce the execution time of tasks, Wang et al. [5] proposed a task-aware service reorganization approach based on quality of service and considered unpredictable situations such as urgent task requirements. Wang et al. [3] proposed a dynamic service composition reconfiguration (DSCRWECPC) approach considering actual constraints and established an optimization algorithm based on Pareto strategy, which solved dynamic uncertainty problem such as equipment failure [4]. When a service exception occurs, the global dynamic scheduling of the service composition is executed from each exception point. Zhang et al. [20] proposed a multi-task-oriented manufacturing service composition (MMSC) model that considers multiple tasks in an uncertain environment to solve uncertainty problems such as urgent tasks and delivery delays; a hyper-heuristic algorithm was proposed to obtain the optimization scheme of the manufacturing service composition. Liu et al. [6] proposed a CMfg dynamic scheduling model that considers dynamic task arrivals. In the model, the failure types and causes of exception conditions faced by cloud services are considered for updating programs and rescheduling production.

Although the above dynamic scheduling uses different models and approaches, the results of the scheduling are all global adjustments to the initial CMfg-SC. Therefore, the service composition optimization based on dynamic scheduling is suitable for CMfg platforms with high intelligence. The cloud service resources on the platform have high data communication and real-time response abilities, and the platform can quickly switch tasks. For most CMfg platforms with decentralized control and low degree of intelligence, frequent task changes will bring significant management difficulties to service providers and lead to poor practical operability.

There are various evaluation indicators currently used in CMfg-SC scheduling; most of them evaluate service composition based on quality of service. Based on the CMfg background, Laili et al. [21] used four second-level indexes (processing time, processing cost, service provider idle rate and delay adaptability) as evaluation indexes of cloud service composition. Yang et al. [22] used six second-level indexes (importance, supply and demand, cost, remaining time, reputation and predetermined cost) as indexes for the service composition evaluation. Based on the evaluation of cloud service composition reputation (CSCR), Xie et al. [23] took two types of stability and collaboration ability as the first-level evaluation indexes of the service composition and three types (execution time, cost and reliability) as the second-level indexes. Li et al. [24] proposed six indexes (reliability, reputation, combination collaboration, combination complexity, execution time and execution cost) to evaluate service composition. From the literature review above, CMfg-SC scheduling is based on three attribute indexes of quality of service (time, cost, reliability) as the basic research [25]. Therefore, this paper combines and evaluates these three attribute indexes, so that the overall quality of service value can be optimized to meet the needs of users. These three attribute indexes can be described as follows [26]: (1) time – from the time the user submits the task to the end of the execution; (2) cost – total cost that the user pays throughout the execution of the task; (3) reliability – ability to successfully execute manufacturing tasks under a given time and condition. Based on the existing research work, this paper considers the flexibility of service as another evaluation index of cloud manufacturing services to measure the adaptability and stability of cloud manufacturing service systems in a dynamic service environment.

2.2. Simulation of CMfg-SC rescheduling strategy

Simulation plays a key role in the design, improvement and evaluation of manufacturing systems. Particularly, digital twin technology can quickly evaluate the operation of the actual system and assist decision-making based on dynamic simulation [27]. The CMfg-SC rescheduling strategy is not to execute global optimization calculations but to establish certain adaptive rules to deal locally with system disturbances or exceptional situations. The simulation of the CMfg-SC rescheduling strategy is to model and run the manufacturing system based on the real environment, while simulating the occurrence of exceptional situations and executing different rescheduling strategies. Then, strategy selection is made by evaluating the performances of different rescheduling strategies. Vijayan et al. [28] designed three scenarios where production resources are interrupted due to exceptions and designed alternative paths of interruptions for each scenario. Simulations of different scenarios show that there are significant differences in system performance when different alternative paths are used. A study by Psarommatis et al. [29] introduces performance indicators for five factors influencing production interruptions and designed a production rearrangement scheme for each factor. The impact of rearrangement production on production quality is quantitatively analyzed by comparing and discussing the results of simulation experiments in the manufacturing workshop. Champati et al. [30] proposed a Greedy-One-Restart (GOR) algorithm that estimates the processing time when canceling and rescheduling CMfg tasks and compared the scheduling performance of the improved algorithm with other algorithms through simulation. It should be noted that rescheduling strategy simulation is aimed at the job shop production environment, and there is scant research on the CMfg platform.

So far, the models used in CMfg system simulation include the discrete event dynamic simulation model [31], multi-agent simulation model and hybrid simulation model [7], among others. Zhao et al. [32] designed a manufacturing simulation platform for the transaction process of enterprises in the cloud environment. The enterprise behavior is described by encapsulating each enterprise into a multi-service agent (Service Agent), and the feasibility of the platform is verified through practical cases. Zhou et al. [33] constructed a multi-agent model based on the CMfg network and designed three different production modes that consider dynamic service environments. The relationship between the production mode and manufacturing is analyzed through simulation experiments. Zhao et al. [34] proposed a multi-agent model and architecture for CMfg simulation based on the concept of service agents, which analyzed the interaction between agents and dynamic environments and the processing mechanism within agents [35]. Self-organizing networks are formed through service agent-driven services that simulate service transactions and collaborations. It can be seen from the above research that multi-agent simulation is very suitable for describing the uncertainty, distribution and dynamics of CMfg, and it thus has become the mainstream approach for CMfg simulation analysis.

To perform the simulation research of the task rescheduling strategy in CMfg mode, this paper uses a multi-agent modeling approach to construct a simulation model of the CMfg service process. Moreover, manufacturing environments with different degrees of disturbance and different rescheduling strategies are designed, and rescheduling strategies are compared and evaluated under different disturbance degrees. Based on existing studies of the cloud service platform, two rescheduling strategies are proposed: (1) Consider resource substitution within the service provider and transfer exceptional tasks to similar resources of the same service provider; (2) consider resource substitution between service providers and transfer exceptional tasks to other service providers with resources. In this paper, we aim to research and evaluate the performances of different rescheduling strategies in different manufacturing environments through simulation, which may help a CMfg platform to adopt appropriate dynamic task rescheduling strategies and reduce losses caused by task exceptions.

3. Results evaluation indexes for rescheduling strategies

3.1. Description of the parameters

In the CMfg platform, users issue service orders to the platform, where each order is divided into several manufacturing tasks, and service providers on the platform provide service resources for the manufacturing tasks. When there is a task exception caused by disturbances, the platform adopts a local rescheduling strategy. This research aims to help the platform to make the decision of rescheduling strategy through simulation evaluation. Two evaluation indexes of quality of service (QoS) and flexibility of service (FoS) are used. The evaluation index QoS is weighted by time, cost and reliability [3], based on most CMfg-SC optimization studies [4]. In addition, since this study pays special attention to the adaptability of rescheduling strategies to different degrees of disturbance, the FoS is added as an evaluation index [36]. To calculate the evaluation indexes, the following definitions must be introduced.

There are N_s types of manufacturing cloud services in the CMfg service system, and each type of service has matching service resources to perform specific manufacturing functions, i.e., $MF = \{mf_j | 1 \le j \le N_s\}$. Services M are supplied by providers: $MS = \{MS_m | 1 \le m \le M\}$. MS_m provides n_m $(1 \le n_m \le N_s)$ types of manufacturing services $S = \{cs_{m,j} | 1 \le j \le n_m\}$, where $cs_{m,j}$ can be described as follows:

$$cs_{m,j} = \{t_{m,j}, a_{m,j}, c_{m,j}, e_{i,j}, rel_m^i\} \tag{1}$$

where $t_{m,j}$ represents the task type of $cs_{m,j}$, $a_{m,j}$ is the amount of resources corresponding to $cs_{m,j}$, $c_{m,j}^i$ is the cost of using task st^i of $cs_{m,j}$ for unit time, and $e_{i,j}$ and rel_m^i represent resource matching rate and reliability of all services provided by MS_m.

To evaluate the three indexes of time, cost and reliability of the CMfg platform, this study uses the following parameters: time, cost and reliability.

Table 1. Parameters of the evaluation indexes.

	Nomenclature		
		MS	Service providers
		n_m	Number of types of resource provided by MS
a_u	Unit amount of task st^i	N	Total number of tasks
$a_{m,j}$	Amount of resource associated with $cs_{m,j}$	N_s	Number of resource of manufacturing services in the entire cloud manufacturing system
$c_{m,j}^i$	Task cost unit time of task st^i of $cs_{m,j}$ of matching resource j	$Norm$	Normalized value of ith index

$cs_{m,j}$	Cloud resource j offered by MS_m	rel	Reliability of all services provided by MS_m
$csc_{m,j}^{i}$	Cost for $cs_{m,j}$ to task st^i	rel_m^i	Reliability of ith task st^i
$cst_{m,j}^{i}$	Time for $ms_{m,j}$ to task st_i	st^i	ith task
CV_x	Coefficient of variation of index x	sst^i	Time required for task st^i using unit amount of benchmark resource
$d^{i,i+1}$	Logistics distance between MS_m undertaking tasks st^i and st^{i+1}	SC	Total cost of service
$dlc^{i,i+1}$	Cost of transporting for unit distance	SN_a^i	Amount of service composition mobilized by MF
$dlt^{i,i+1}$	Logistics time per unit distance		
$e_{i,j}$	Matching rate of task st^i using resource j	SN_n^i	Number of normal responses for task st^i
FL	Comprehensive fluctuation value	ST	Total time of service
$lc^{i,i+1}$	Logistics cost between adjacent tasks st^i and st^{i+1}	$t_{m,j}$	Task type of $cs_{m,j}$
		x_i	ith index
$lt^{i,i+1}$	Logistics time between two MS providing services to adjacent subtasks	$\bar{x}$	Mean of index x
		$\delta^{i,i+1}$	A Boolean variable characterizing whether logistics between st^i and st^{i+1} exists
mf_j	jth manufacturing function		
M	Number of MS	ω_i	Weight value of the ith index
MF	Set of manufacturing functions	σ_x	Standard deviation of index x

3.2. Calculation of QoS

Based on previous research [6], QoS is defined as an evaluation index to measure the CMfg-SC metrics including time, cost and reliability. This section proposes the approaches to calculate time, cost and reliability.

3.2.1. Time

For CMfg-SC with consideration of service and logistics, the total service time ST includes both task time $cst_{m,j}^{i}$ and logistics time $lt^{i,i+1}$. Task time $cst_{m,j}^{i}$ can be calculated as follows:

$$cst_{m,j}^{i} = (sst^i \times a_u)/e_{i,j} \tag{2}$$

where sst^i is service time of using unit task amount of task st^i, a_u is the volume for task st^i, and $e_{i,j}$ is the matching rate between task st^i and resource j.

Logistics time $lt^{i,i+1}$ between st^i and st^{i+1} can be calculated as follows:

$$lt^{i,i+1} = \delta^{i,i+1} \times dlt^{i,i+1} \times d^{i,i+1} \tag{3}$$

where $\delta^{i,i+1}$ and $d^{i,i+1}$ represent the Boolean variable and distance (km) between providers undertaking st^i and st^{i+1}, and $dlt^{i,i+1}$ is logistics time per unit distance.

The total service time ST can be calculated as follows:

$$ST = \sum_{i=1}^{N}\left(cst_{m,j}^{i} + lt^{i,i+1}\right) \tag{4}$$

where N is the total number of tasks in MF.

3.2.2. *Cost*

For CMfg-SC with consideration of service and logistics, the total service cost SC includes both task cost $csc_{m,j}^{i}$ and logistics cost $lc^{i,i+1}$. Service cost $csc_{m,j}^{i}$ of $cs_{m,j}$ can be calculated as follows:

$$csc_{m,j}^{i} = c_{m,j}^{i} \times cst_{m,j}^{i} \tag{5}$$

where $c_{m,j}^{i}$ is the task cost of st^{i} for unit time that matches resource j.

Logistics time $lc^{i,i+1}$ between st^{i} and st^{i+1} can be calculated as follows:

$$lc^{i,i+1} = \delta^{i,i+1} \times dlc^{i,i+1} \times d^{i,i+1} \tag{6}$$

where $\delta^{i,i+1}$ and $d^{i,i+1}$ represent the Boolean variable and distance (km) between providers undertaking st^{i} and st^{i+1}, and $dlc^{i,i+1}$ is logistics time per unit distance.

The total service cost SC can be calculated as follows:

$$SC = \sum_{i=1}^{N} \left(csc_{m,j}^{i} + lc^{i,i+1} \right) \tag{7}$$

where N is the total number of tasks in MF.

3.2.3. *Reliability*

For CMfg-SC with consideration of background, the reliability can be calculated as follows:

$$rel = \prod_{i=1}^{N_s} rel_{m}^{i} \tag{8}$$

where rel_{m}^{i} is the reliability of the ith service task, which represents the ability of CMfg-SC to operate normally (no exceptional tasks). Use the ratio of the number of normal responses SN_{n}^{i} to the total number of called tasks SN_{a}^{i} to expressed it in a service cycle, i.e.,

$$rel_{m}^{i} = SN_{n}^{i} / SN_{a}^{i} \,. \tag{9}$$

Because indexes of time and cost fall into different ranges and have different units, they need to be normalized to a range between 0 and 1 [37] for the convenience of calculations.

For a negative index like service time, it is normalized as follows:

$$Norm(ST) = \begin{cases} \frac{\max ST - ST}{\max ST - \min ST}\,, & \min ST \neq \max ST \\ 1\,, & \min ST = \max ST \end{cases} \tag{10}$$

where $maxST$ and $minST$ represent the maximum and minimum values of index aggregation values of ST in all the possible combined paths. After normalization, all values of indexes will be within the range of $[0,1]$.

For a negative index like service cost, it is normalized as follows:

$$Norm(SC) = \begin{cases} \frac{\max SC - SC}{\max SC - \min SC}, & \min SC \neq \max SC \\ 1, & \min SC = \max SC \end{cases} \tag{11}$$

where $maxSC$ and $minSC$ represent the maximum and minimum values of index aggregation values of SC in all the possible combined paths. After normalization, all index values will be within the range of $[0,1]$. Because service reliability is within the range of $[0,1]$, normalization is not required for this index.

To weight the normalized indexes in a simple manner, the maximum performance value $\mathrm{Max}(QoS)$ of QoS can be calculated as follows [38]:

$$\mathrm{Max}(QoS) = \omega_1 Norm(ST) + \omega_2 Norm(SC) + \omega_3 rel \tag{12}$$

$$\sum_{i=1}^{3} \omega_i = 1 \tag{13}$$

where $Norm(ST)$ and $Norm(SC)$ represent the normalized values of the time and cost index attribute, and ω_i represents the weight value of the ith indicator, which is selected according to the user's evaluation index preference ($\omega_i \in [0,1]$). In this study it is assumed that users have the same preference and set the three indexes ω_i to one third [13].

3.3. Calculation of FoS

When changing the disturbance degree of the manufacturing environment, the adaptability of the same task rescheduling strategy may have obvious deviation. It is possible that a certain rescheduling strategy performs very well in a stable manufacturing environment but becomes inadequate when the disturbance of the manufacturing environment is significant. When comparing various rescheduling strategies, in addition to paying attention to QoS index, it is also necessary to consider the adaptability of different degrees of environmental disturbance, i.e., flexibility of service (FoS).

This study defines FoS as the degree of comprehensive fluctuation of time, cost and reliability after adopting a certain task rescheduling strategy under different degrees of disturbance. The fluctuation degree FL_x is calculated using the coefficient of variation of the index x at various degrees of disturbance, according to the following formula:

$$FL_x = \sqrt{\sum_{i=1}^{J} (x_i - \bar{x})^2 / (J-1)} / \bar{x} \tag{14}$$

where x_i represents the value of a certain index under the degree of environmental disturbance i ($i \in [1, J]$), J is the amount of environmental disturbance degrees, and $\bar{x}$ represents the mean value of a certain index under all disturbance degrees. This study sets the degree of disturbance of small, medium and large, i.e., $J = 3$.

Set the weight coefficients for the disturbance degrees of the three indicators to be ω_1, ω_2 and ω_3, respectively, and the comprehensive fluctuation degree FL can be calculated as follows:

$$FL = \omega_1 FL_{ST} + \omega_2 FL_{SC} + \omega_3 FL_{rel} \tag{15}$$

The higher the comprehensive degree of fluctuation is, the more unstable the CMfg service platform is in response to external disturbance and the lower the FoS, which can be calculated according to the following formula:

$$FoS = 1 - FL \tag{16}$$

4. Multi-agent model of service processes in CMfg mode

4.1. Conceptual model of CMfg service process

The CMfg service platform $\boldsymbol{P}$ has $\boldsymbol{M}$ providers $\boldsymbol{MS_m}$ and provides $\boldsymbol{N_s}$ diverse types of manufacturing services. The user publishes the orders to the CMfg platform, which then processes these orders $\boldsymbol{O_i(i = 1, 2, \ldots, n)}$ into different tasks, where $\boldsymbol{Task(N)} = \{\boldsymbol{T_1}, \boldsymbol{T_2}, \ldots, \boldsymbol{T_N}\}$ represents the task pool. Each task requires one or more services, and the platform configures the cloud service resource $\boldsymbol{R_n}$ for each task. The process is shown in Figure 1.

4.2. Agent model of CMfg service

Due to the serviceability and autonomy of service agents, they can actively and spontaneously conduct services and cooperation in the simulation model. The agents can achieve their own functions and purposes through certain rules and strategies in CMfg service system. Based on the conceptual model, this study extracts six types of agent models: communication agents, task agents, resource agents, scheduling agents, order agents. and user agents. These six types of agent models are described below.

4.2.1. Communication agent model

The service information between several types of agent interfaces is conveyed through the communication agent, including receiving and sending of information. The cloud-made communication agent model can be described as follows:

$$Msg_{SA} = PortCode.send(new\ Message(), Portn) \tag{17}$$

where $PortCode$ represents the interface for sending information;
$Portn$ represents the interface for receiving information;
$new\ Message()$ is a communication body between interfaces and carries the complete information content, as defined by the following formula:

$$new\ Message() = < MSGID, Msg_{type}, Msg_{amount} > \tag{18}$$

where $MSGID$ represents the unique identifier of communication information. Msg_{type} represents the type of communication body, and Msg_{amount} represents the number of communication bodies.

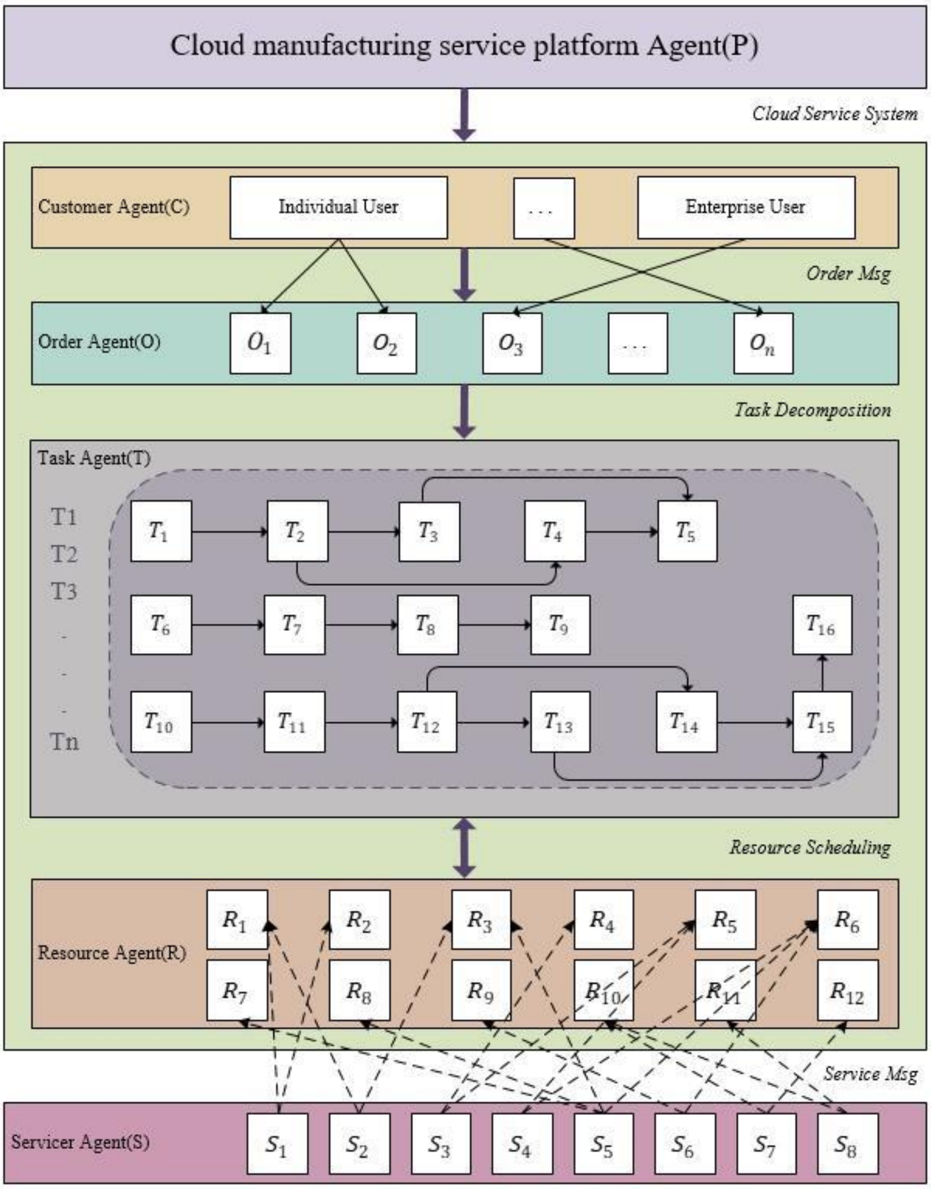


Figure 1. Conceptual model of CMfg service process.

4.2.2. Task agent model

The task agent is an agent that accepts task messages and achieves task execution functions. The agent model can be described as follows:

$$TS_{cmfg} = < TSID_{flu}, State_{TS}, Amount, Time_{TS}, Clk_{TS}, RS_{TS}, Order_{cmfg}, Queue_{TS} > \tag{19}$$

where $TSID_{flu}$ is the unique identifier of the task agent, which is used to determine the task information from different disturbance environments;

$State_{Ts}$ stands for state information for task agent, including status such as publishing, waiting, transferring and executing;

$Amount$ represents the number of service tasks carried by the task agent;

$Time_{TS}$ is the execution time of the service task;

Clk_{TS} is a clock of a task agent which records a task assignment when triggered;

RS_{TS} represents the resource matched by the service task, and the matching rates between different tasks and resources are different;
$Order_{cmfg}$ represents the order agent to which the service task belongs.

$$Order_{cmfg} =< Type_O, Amount_O, Clk_{OS}, Queue_O > \tag{20}$$

In Eq. (20), $Type_O$ is the order type; $Amount_O$ is the order number; Clk_{OS} is a clock of the order agent, which records an order assignment; $Queue_O$ is the order queue in the order agent and carries the order sequence that arrives in real time; $Queue_{TS}$ is the message queue in the task agent and carries the task content and sequence that arrive in real time.

4.2.3. Resource agent model

Due to the diversification of CMfg service resources, CMfg resources are intelligently packaged, and the resource pool is connected digitally. The resource model of the service agent is described as follows:

$$RS_{cmfg} =< RSID, Info_{state}, Templ, Data, SARS, Func_{01}, Func_{02}, \dots, Func_n > \tag{21}$$

where $RSID$ is the unique identifier of the service resource;
$Info_{State}$ is the state of the resource, such as $Idle$, $Busy$, $Suspend$, etc., and can define different real-time states of the resource;
$Templ$ represents a resource template, which is divided into static and dynamic sections, which can describe virtual resources by metadata, such as static resource information and dynamic data;
$Data$ represents the data recorded on static and dynamic resource templates, used to collect, extract and process for service resources;
$SARS$ represents the service agent to which the resource belongs; and
$Func_n$ is the function of virtualized resources used to encapsulate the various resources.

4.2.4. Scheduling agent model

The scheduling agent has service-oriented autonomy and can simulate the collaborative behavior between users and service providers. At the same time, it can autonomously interact with information data, simulate real-time pattern demonstration, scheduling and network evolution and execute four main behaviors: 1) publish orders and wait for recommended matching service providers to cooperate, 2) process service requirements from service providers, 3) schedule service tasks dynamically according to service strategies and 4) respond to the instructions of the cloud service platform according to the current status information.

According to the above agent behaviors, the scheduling agent model is described as follows:

$$\begin{aligned} SA_{cmfg} =< & SAID_{flu}, Info_{state}, Info_{basic}, Msg_{SA}, Clk_{SA}, Func_{req}, \\ & Func_{trans}, Func_{strategy}, Func_{query}, Func_{respond} > \end{aligned} \tag{22}$$

where $SAID_{flu}$ is the unique identifier of encapsulated scheduling agent used to determine service agents in different environments;
$Info_{state}$ is the state of information used to describe service requirement, publish a service request, make service selection and respond to service. These four states correspond to feedback of four functions as follows:

$$Info_{state} = < State_{req}, State_{trans}, State_{query}, State_{respond} > \tag{23}$$

where $Info_{basic}$ is the basic information of the scheduling agent and stores the description, attributes, parameters, rules and other agent's information. The model is described as follows:

$$Info_{basic} = < DATA_{basic}, DATA_{rule}, DATA_{type}, DATA_{collection}, DATA_{respond} > \tag{24}$$

where Clk_{SA} is the clock of the scheduling agent used to record task, service time, service efficiency and other agent's information;

$Func_{req}$ comprises the function of service requirement used to execute and update the agent's basic information;

$Func_{trans}$ is the process function triggered when provider requests service;

$Func_{strategy}$ is the function of the service agent to execute the rescheduling strategy when the task is exceptional;

$Func_{query}$ is a function of the command query;

$Func_{respond}$ is the response function that updates the basic data of the matching service provider.

4.3. Simulation model of CMfg service

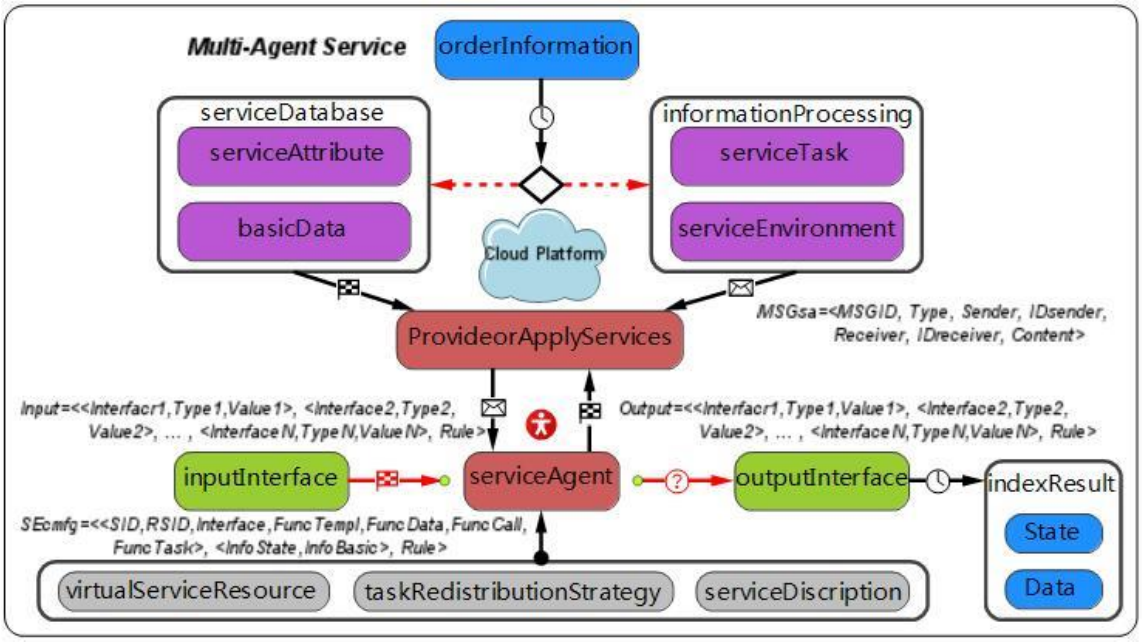


Figure 2. Internal structure model of CMfg service agent.

In this study, all the experiments were performed in Anylogic4.8 and implemented in a PC with an Intel i7-9100 U, 3.6 GHz, with 8 GB RAM which uses the operating system Windows 10 (64 bit) and Java language for secondary development.

The communication of the service agent is performed based on the service protocol. The $Content$ is the message content, which is composed of the autonomy of the agent $Action$ and the message body Msg, as shown in Figure 2.

In the production environment with the CMfg mode, $Action$ is the function of a message, and Msg is the message body provided by an object, which is a pair of key and value and contains the type of message and the task data.

The internal structure of the service agent is based on the architecture described in Figure 2. This paper proposes a simulation modeling approach of a multi-agent, and the detailed process flowchart is shown in Figure 3.

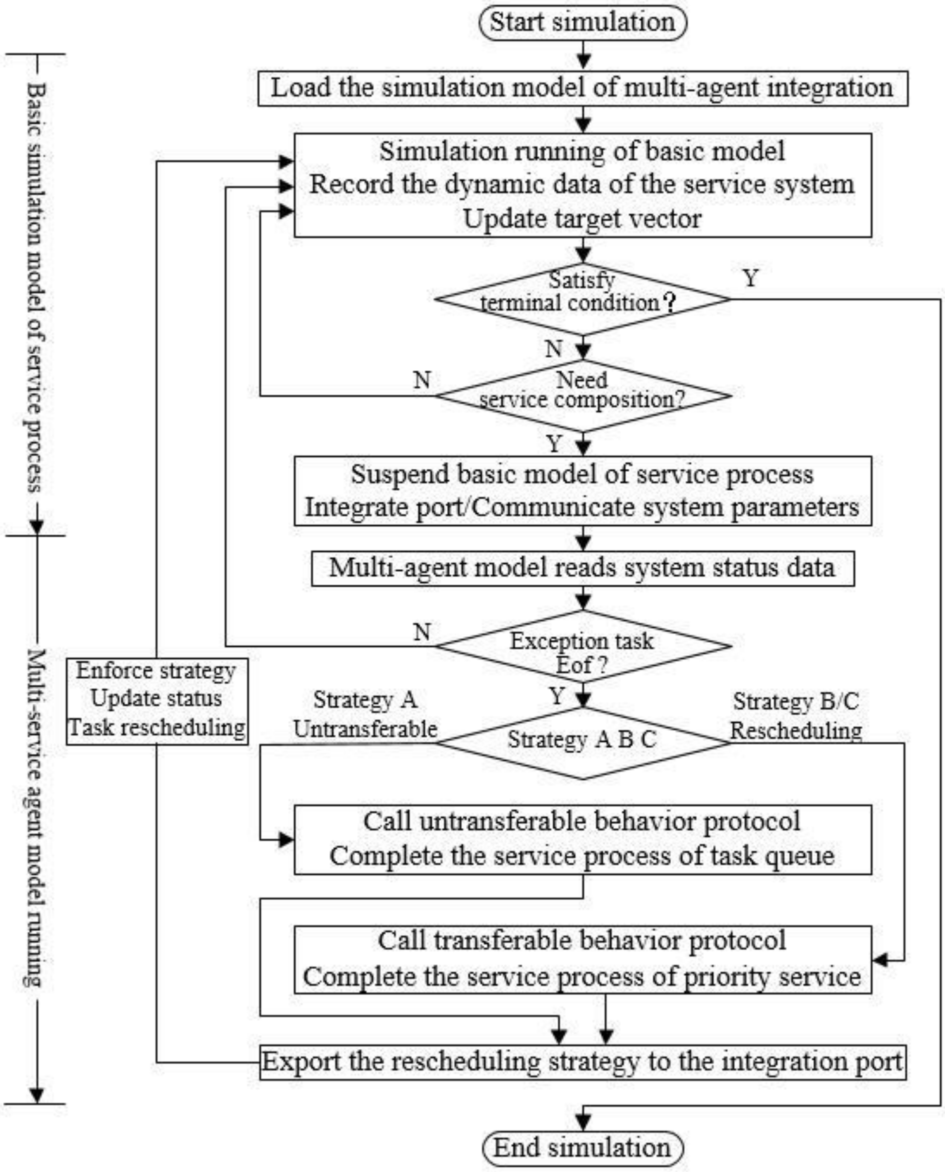


Figure 3. Flow chart of simulation experiment.

Step 1: Start simulation, and the CMfg platform interprets the service task and loads the simulation model of multi-agent.

Step 2: Run the basic simulation model, record the dynamic data of each service task in the service process, and update the target vector in real time. Meanwhile, start to run the *StateChart* module shown in Figure 4.

Step 3: Estimate whether the service task has completed. If so, end the simulation. Otherwise, proceed to the next step.

Step 4: Estimate whether the service task requires service composition [39]. If so, proceed to the next step and run the end module in Figure 4. Otherwise, go back to step 2.

Step 5: Communicate the state data of the system through the integrated interface of the multi-service agent model.

Step 6: Determine whether the service task is abnormal and run the *TaskException* module shown in Figure 4. If so, execute the task without transferring strategy or the task rescheduling strategy, and run the *TaskRescheduling* module in Figure 4. Otherwise, go back to step 2.

Step 7: Output the rescheduling strategy to the integration interface and execute it. Then, update the task status and execute task rescheduling and logistics. Finally, return to step 2.

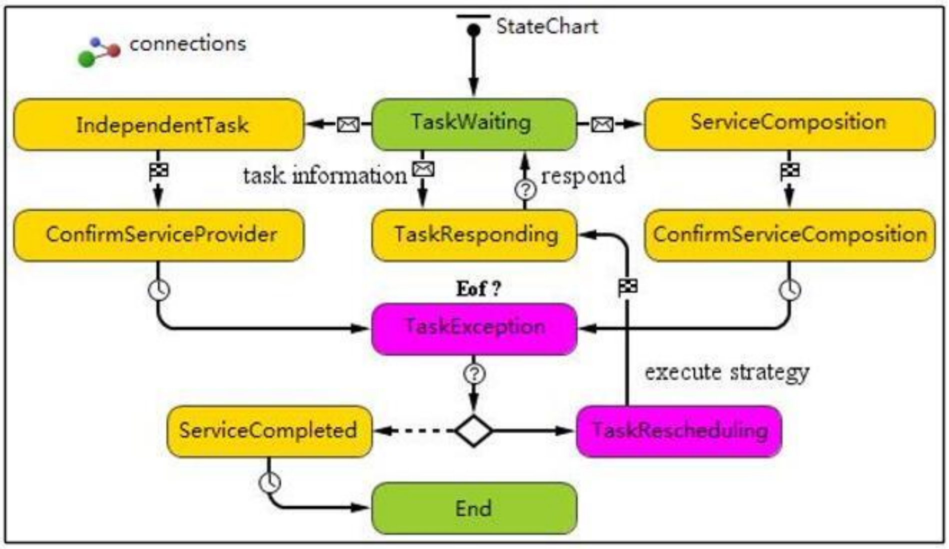


Figure 4. Technology architecture of simulation experiment.

5. Experiment plan

5.1. Case simulation

This paper takes the CMfg service of a complex electronic product as an example to illustrate how to simulate and evaluate the rescheduling strategy of exceptional tasks. In this case study, there are three kinds of order request information Msg_{Q_i} (where $i \in [1,3]$), and eight service providers ($S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8$) execute 16 different types of CMfg service tasks, which require 12 types of service resources.

$Queue_{order}(O_3, O_1, O_3, O_2, O_3, O_2, O_1, O_2, O_1, O_2)$ is the sequence of order for the service cycle, where the corresponding tasks of order O_1 are $(T_1, T_2, T_3, T_4, T_5)$, the tasks of order O_2 are (T_6, T_7, T_8, T_9), the tasks of order O_3 are $(T_{10}, T_{11}, T_{12}, T_{13}, T_{14}, T_{15}, T_{16})$, and the specific task relationships in the three orders are shown in Figure 5.

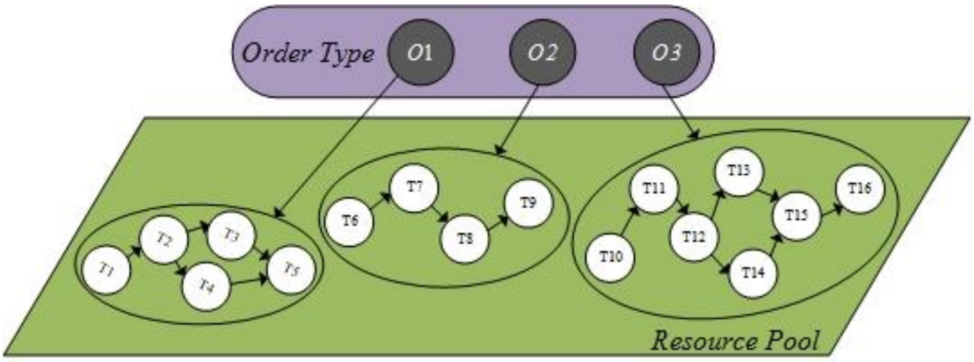


Figure 5. Task relationships of three orders.

The task resource relationships and service times of the orders are shown in Table 2.

Table 2. Service times of tasks in the CMfg platform.

	Task service time (min)															
Resource	R_1	R_1	R_2	R_3	R_1	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}	R_9	R_{12}
Task	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}	T_{16}
S_1	30	18	37													
S_2				20	41	52										
S_3							31									
S_4								19								
S_5									11	23	12					
S_6												24			25	
S_7													13			28
S_8														19		

In this case, different orders have different unit service costs and logistics costs, as shown in Table 3.

Table 3. Cost information of orders.

Price (dollar/min)	Order O_1	Order O_2	Order O_3
Service cost	4	5	3
Logistics cost	2.5	3.5	1.5

The degree of disturbance in this case is described by the resource failure rate, the order urgent request rate and the logistics interruption rate. In the service process of cloud manufacturing, the higher the resource failure rate is, the more frequent the urgent demand for orders and the more frequent the interruptions in the logistics process, indicating that the disturbance degree of the service environment is greater. According to different disturbance levels, this paper divides the disturbance degree of the service environment into three types, i.e., small disturbance, medium disturbance and large disturbance. In the simulation, different disturbance degrees are described by setting the probability of each disturbance scenario, as shown in Table 4.

Table 4. Setting of environment fluctuations.

Condition/Fluctuation (rate)	Resource failure	Order insert	Transportation interrupt
Small disturbance	5%	10%	5%
Medium disturbance	10%	15%	10%
Large disturbance	15%	20%	15%

5.2. Rescheduling strategies of considering resource substitution

The CMfg platform integrates rich manufacturing resources through network service, and various resources have high flexibility and large substitutional space, which improve possibilities for the rescheduling strategies of exceptional tasks.

Therefore, in this case study, resource substitution is considered, and the task rescheduling strategies of internal and external transfer of service provider are proposed. In the simulation experiment, the strategy of maintaining the original scheduling scheme without rescheduling is named *Strategy A*, and the strategies of internal and external transfer of service provider are named *Strategy B* and *Strategy C*, respectively. These three strategies are defined as follows.

Strategy A: Do not transfer exceptional service tasks. According to the initial settings in Table 2, queue up for service on a first-come, first-served basis.

Strategy B: Transfer exceptional service tasks within the service provider. The scheduling agent searches for substitutional resources for the exceptional task within the service provider and transfers the task to an alternate resource that is idle and has the highest matching degree. Based on this strategy, the time for task blocking will be shortened, and logistics costs will be negligible due to the internal transfer of service provider. However, due to the use of substitutional resource, the matching degree of the task with the resource is reduced, and the processing time of the task is extended. The substitutional resources for Strategy B and the settings for matching degree are shown in Table 5.

Table 5. Substitution resource matching data of stack tasks.

Task-Resource	R_{1-2}	R_{1-3}	R_{2-3}	R_{4-6}	R_{5-6}	R_{6-7}	R_{6-8}	R_{7-8}	R_{9-10}	R_{9-11}	R_{9-12}	R_{10-12}
Matching rate (%)	80	75	70	75	75	55	65	80	75	75	70	80

Strategy C: Transfer exceptional service tasks outside the service provider. The scheduling agent searches for substitutional resources for the exceptional task outside the service provider and transfers the task to the substitutional resource of the other provider that is idle and has the highest matching degree to the original resource. Based on this strategy, the time for task blocking will be minimized. Since the resource search scope is expanded to all service providers, the matching degree between task and resource is high, and the task service time will not be affected. However, the change of service provider will lead to additional logistics time and logistics cost. In this case study, the logistics distances (km) between the service providers are shown in the following triangular matrix SD_{ij}.

$$SD_{ij} = \begin{bmatrix} 0 & 16 & 92 & 98 & 127 & 124 & 156 & 164 \\ & 0 & 77 & 84 & 112 & 107 & 140 & 148 \\ & & 0 & 20 & 36 & 44 & 68 & 84 \\ & & & 0 & 31 & 59 & 77 & 97 \\ & & & & 0 & 50 & 53 & 78 \\ & & & & & 0 & 34 & 42 \\ & & & & & & 0 & 27 \\ & & & & & & & 0 \end{bmatrix} \quad (25)$$

where i and j represent different service providers $(i, j \in [1, 8])$.

5.3. Experiment scheme

For the three disturbance degrees set in Table 4, nine simulation experiments were executed on the three rescheduling strategies A, B and C, and the performances of different strategies under different disturbance degrees were evaluated. During the simulation, when the blocking time of the task to be served exceeds 10% of the execution time of the task, it is marked as a task exception. When a task exception occurs, the exceptional task is scheduled by the agent according to the preset rescheduling strategy. For preset order sequences, we run ten simulation experiments each time; we calculate the mean values of time, cost and reliability index after 10 simulations and further calculate the evaluation index QoS [40] and FoS.

6. Experiment results and sensitivity analysis

6.1. Results analysis and discussion

Table 6 shows the QoS index values of each strategy under different disturbance degrees obtained based on the simulation output data and the calculation approach of the evaluation index QoS in Section 3.2.

Table 6. QoS values of the three strategies under each disturbance level.

Strategy	Strategy A			Strategy B			Strategy C		
Environment	ST	SC	rel	ST	SC	rel	ST	SC	rel
Small disturbance	0.6899	0.5361	0.6047	0.6878	0.6746	0.6429	0.7523	0.7548	0.6952
Medium disturbance	0.4777	0.4747	0.5601	0.5322	0.6404	0.5069	0.6904	0.6922	0.6663
Large disturbance	0.3938	0.3706	0.4262	0.441	0.5268	0.488	0.632	0.6172	0.63
QoS	$QoS_A^{Small} = 0.6102$ $QoS_A^{Medium} = 0.5041$ $QoS_A^{Large} = 0.3969$			$QoS_B^{Small} = 0.6684$ $QoS_B^{Medium} = 0.5598$ $QoS_B^{Large} = 0.4853$			$QoS_C^{Small} = 0.7341$ $QoS_C^{Medium} = 0.683$ $QoS_C^{Large} = 0.6264$		

From Table 6 it can be noticed that with the increase of the environmental disturbance degree, the values of each index show a decreasing trend. As can be seen from Figure 6, the QoS index values of strategies A and B are always significantly smaller than strategy C, indicating that strategy C has advantages in all three disturbance levels. Second, strategy A performs worst of all disturbance levels, indicating that rescheduling of exceptional tasks in CMfg is necessary.

In Figure 7 the performances of different indicators are compared; strategy C has the obvious advantages in the time index and the worst performance as the reliability index. There is no difference in the time indexes of strategies A and B at small disturbance, while at medium disturbance, the reliability index of strategy A is better than that of strategy B. Therefore, the CMfg service platform can choose different service strategies based on the results of the simulation evaluation.

Based on the data in Table 7 above and the calculation approach of the evaluation index FoS in Section 3.3, the FoS index value of each strategy is achieved, as shown in Table 7.

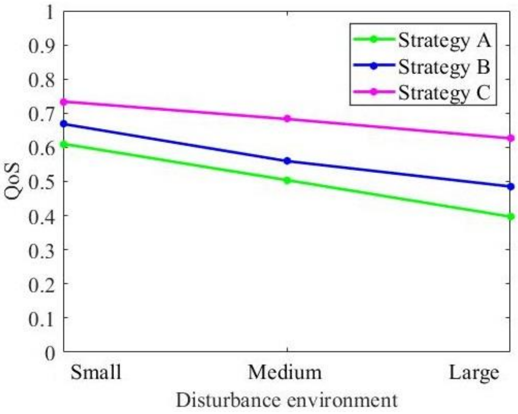
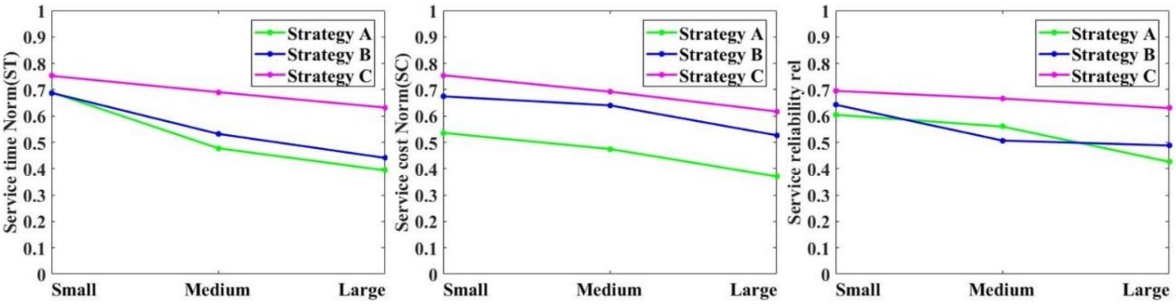


Figure 6. QoS of three strategies under different disturbance environments. The curves of different disturbances are obtained by averaging over ten times of simulating.



a. Time-Disturbance level

b. Cost-Disturbance level

c. Reliability-Disturbance

Figure 7. Performance values of three indexes based on three strategies.

Table 7. FoS values of the three strategies under each disturbance degree.

Strategy	Strategy A			Strategy B			Strategy C		
FL_x	FL_{ST}	FL_{SC}	FL_{rel}	FL_{ST}	FL_{SC}	FL_{rel}	FL_{ST}	FL_{SC}	FL_{rel}
FL	0.2932	0.1817	0.1752	0.2254	0.126	0.1548	0.087	0.1001	0.0492
FoS_x	0.7068	0.8183	0.8248	0.7746	0.874	0.8452	0.913	0.8999	0.9508
FoS	$FoS_A = 0.7833$			$FoS_B = 0.8313$			$FoS_C = 0.9212$		

From Table 7 and experimental results it can be noticed that the FoS of strategy C is the highest, followed by strategy B, and that of strategy A is the lowest. In this case, the most stable performance can be achieved by transferring the exceptional task to an external service provider. The FoS performances of different indicators can be explored further, as shown in Figure 8.

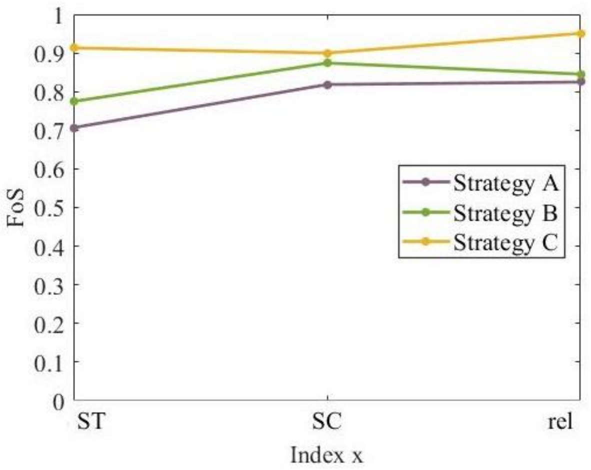


Figure 8. FoS of three strategies under different indexes. The curves of different indexes are obtained by averaging over ten times of simulating.

Figure 8 shows that the service flexibility of strategy C in the three indexes of time, cost and reliability is always greater than that of strategy A and B. However, the situations in the flexibility of the three indicators are not the same. Strategy C has obvious advantages in time flexibility and reliability flexibility but not in service flexibility.

To further study the influence of the parameters of different rescheduling strategies on the QoS and FoS indexes, the parameter sensitivity analysis of the resource matching rate in strategy B and the logistics distance in strategy C need to be examined.

6.2. Sensitivity analysis experiment

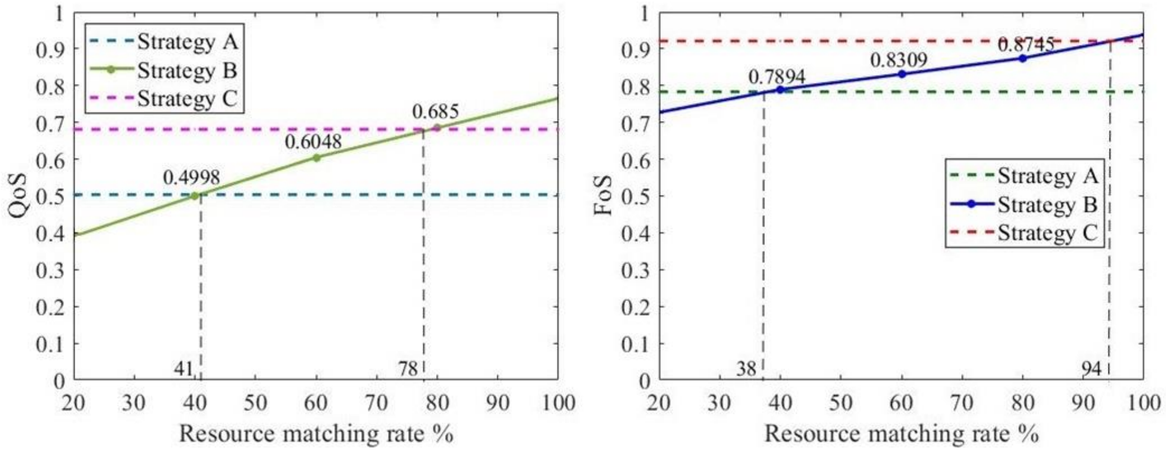
(1) Sensitivity analysis of resource matching rate

In the simulation experiment of strategy B, the data in Table 5 are used as the matching rates of substitutional resources. When a task exception occurs, the scheduling agent searches for the substitute resource with the highest matching rate within the service provider to reschedule the task. In the sensitivity analysis of this section 6.2, the matching rate of substitutional resource is regarded as a variable parameter, which is set to 40%, 60% and 80%, respectively. Then, through simulation experiments, the QoS values under different matching rates are calculated, and the results obtained are shown in Table 8.

Table 8. QoS sensitivity analysis of resource matching rate.

Influence factor	Resource matching rate %		
Variable index	40	60	80
QoS	0.4998	0.6048	0.685

From Table 8 it can be observed that the resource matching rate shows a positive correlation with QoS; in other words, the higher the matching rate is, the larger the QoS. As shown in Figure 9a, when the resource matching rate exceeds 41%, the QoS of strategy B will be better than that of strategy A; moreover, when the resource matching rate exceeds 78%, the QoS of strategy B will be better than that of strategy C. Therefore, when the resource matching rate within the service provider changes in the range $[0, 78\%]$, strategy C is always better than strategy A and B; and when the resource matching rate within the service provider changes in the range $[78\%, 100\%]$, strategy B is always better than strategy A and C. This also shows that if the resource resilience within the service provider is large enough, strategy B will be a better choice; otherwise, strategy C should be selected.



a. QoS

b. FoS

Figure 9. Sensitivity analysis of resource matching rate based on QoS and FoS. There are ten variables for each parameter, and the curves are obtained by taking the average of results of three simulations.

The sensitivity analysis of the evaluation index FoS is executed on the resource matching rate in strategy B, and the results are shown in Table 9.

Table 9. FoS sensitivity analysis of resource matching rate.

Influence factor	Resource matching rate %		
Variable index	40	60	80
FoS	0.7894	0.8309	0.8745

As shown in Table 9, the resource matching rate shows a positive correlation with FoS, i.e., the higher the matching rate is, the larger the FoS. As shown in Figure 9b, the change curve of FoS, when the resource matching rate within the service provider changes in the range $[0, 94\%]$, strategy C is always better than strategy A and B. When the resource matching rate within the service provider changes in the range $[94\%, 100\%]$, strategy B will be better than strategy A and C. However, given that the resource matching rate within the service provider hardly exceeds 94%, the advantage of FoS of strategy C is stable.

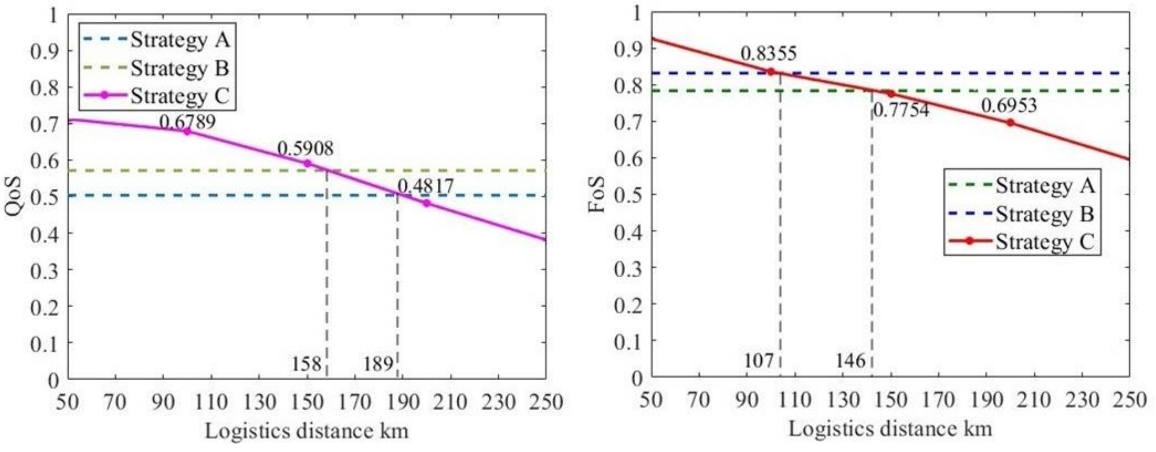
(2) Sensitivity analysis of logistics distance

In the simulation experiment of strategy C, use the data in section 5.2 as the logistics distance of task transfer. When the task exception occurs, search the other service providers for substitutional resources outside the service provider. In the sensitivity analysis of this section, the logistics distance of task transfer is regarded as a variable parameter, which is set to 100, 150 and 200 km, respectively. Then, through simulation experiments, the QoS values under different distances are calculated, and the results obtained are shown in Table 10.

Table 10. QoS sensitivity analysis of logistics distance.

Influence factor	Logistics distance km		
Variable index	100	150	200
QoS	0.6789	0.5908	0.4817

As shown in Table 10, the logistics distance shows a negative correlation with QoS; in other words, the greater the distance is, the smaller the QoS. In Figure 10a, it can be observed that, when the logistics distance outside the service provider changes in the range $[0,158]$, strategy C is always better than strategy A and B; when the logistics distance outside the service provider changes in the range $[158,200]$, strategy B is always better than strategy A and C. This also shows that if the distance of the task transfer can be controlled within 158 km, strategy C maintains the highest QoS. Conversely, strategy C is no longer the optimal rescheduling strategy, and the advantages of strategy B are more obvious.



a. QoS

b. FoS

Figure 10. Sensitivity analysis of logistics distance based on QoS and FoS. There are three variables for each parameter, and the curves are obtained by taking average of results of three simulation.

The sensitivity analysis of the evaluation index FoS is executed on the logistics distance in strategy C, and the results are shown in Table 11.

Table 11. FoS sensitivity analysis of logistics distance.

Influence factor	Logistics distance km		
Variable index	100	150	200
FoS	0.8355	0.7754	0.6953

From Table 11 it can be noticed that the logistics distance shows a negative correlation with FoS; in other words, the greater the matching rate is, the smaller the FoS. As shown in Figure 10b, when the logistics distance outside the service provider changes in the range $[0,107]$, strategy C is always better than strategy A and B; when the logistics distance outside the service provider changes in the range $[107,200]$, strategy B is always better than strategy A and strategy C. This also shows that if the distance of the task transfer can be controlled within 107 km, strategy C will maintain the highest FoS. Otherwise, the FoS of Strategy B will surpass Strategy C.

7. Conclusions

Compared to traditional manufacturing environments, the CMfg environment extends to multi-user agent, multi-service agent and multi-regional spaces, so the process of manufacturing services is exposed to greater uncertainty, which makes it more prone to require exceptional service tasks. In this situation, the rescheduling strategy of service tasks plays a significant role in the QoS and FoS of cloud services. In this paper, based on the multi-agent simulation modeling approach, we simulate and evaluate the service process and task rescheduling strategy of CMfg and analyze the impacts of different task rescheduling strategies on system performance under various system disturbances.

The contributions of this paper are summarized as follows. (1) We not only focus on the QoS index of CMfg but also considered the adaptability of task rescheduling strategies to various system disturbances and proposed the FoS index of CMfg. (2) We consider the substitution of resources and propose internal and external transfer strategies of service provider. (3) We propose a multi-agent simulation model of the cloud service process that can better describe the autonomy and interaction of various types of interference factors and reflect the complex process and uncertain environment of CMfg services. The established model and simulation research approach are close to realistic scenarios, which can provide dynamic and quantitative evaluation of various rescheduling strategies, and it is useful for CMfg platforms to make more rational decisions. (4) The results of the simulation experiments show that the simulations proposed in this paper are able to explore and dynamically evaluate different rescheduling strategies from multiple perspectives, whereas the sensitivity analysis provides a comprehensive basis for rescheduling decisions.

As a fundamental study, this paper only executed simulation and evaluation of two rescheduling strategies (the internal and external transfer of service providers). Future work will refine the rescheduling strategies. More factors such as task importance, resource scarcity and cooperation preference will be considered to develop more flexible rescheduling strategies. From the perspective of disturbance factors, this paper only described the degree of fluctuation caused by three factors, which include resource failure, order change and logistics interruption; more disturbance factors will be considered in the future so that the simulation scenario can be made more realistic.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (No.71871018). The authors gratefully acknowledge the anonymous reviewers for insightful comments that helped us improve the quality of this paper.

Conflict of interest

The authors declare that they have no conflict of interests.

References

1. B. H. Li, L. Zhang, S. L. Wang, F. Tao, J. W. Cao, X. D. Jang, et al., Cloud manufacturing: A new service-oriented networked manufacturing model, *Comput. Integr. Manuf. Syst.,* **16** (2010), 1–7.
2. B. H. Li, L. Zhang, L. Ren, X. D. Cai, F. Tao, Y. L. Luo, et al., Further discussion on cloud manufacturing, *Comput. Integr. Manuf. Syst.,* **17** (2011), 449–457.
3. Y. K. Wang, S. L. Wang, S. Gao, X. X. Guo, B. Yang, Adaptive multi-objective service composition reconfiguration approach considering dynamic practical constraints in cloud manufacturing, *Knowl-Based Syst.,* **234** (2021), 107607. https://doi.org/10.1016/j.knosys.2021.107607
4. Y. K. Wang, S. L. Wang, L. Kang, S. B. Wang, An effective dynamic service composition reconfiguration approach when service exceptions occur in real-life cloud manufacturing, *Robot Comput. Integr. Manuf.,* **71** (2021), 102143. https://doi.org/10.1016/j.rcim.2021.102143
5. Y. Wang, Z. W. Dai, W. Y. Zhang, S. Zhang, Y. B. Xu, Q. Chen, Urgent task-aware cloud manufacturing service composition using two-stage biogeography-based optimisation, *Int. J. Comput. Integ. M,* **31** (2018), 1034–1047. https://doi.org/10.1080/0951192X.2018.1493230
6. Y. K. Liu, H. G. Liang, Y. Y. Xiao, H. F. Zhang, J. X. Zhang, L. Zhang, et al., Logistics-involved service composition in a dynamic cloud manufacturing environment: A DDPG-based approach, *Robot Comput. Integr. Manuf.,* **76** (2022), 102323. https://doi.org/10.1016/j.rcim.2022.102323
7. D. W. Ren, X. D. Zhang, Research on flexibility of production system based on hybrid modeling and simulation, *Math. Biosci. Eng.,* **18** (2021), 933–949. https://doi.org/ 10.3934/mbe.2021049
8. L. Zhang, L. F. Zhou, L. Ren, Y. J. Laili, Modeling and simulation in intelligent manufacturing, *Comput. Ind.,* **112** (2019), 103123. https://doi.org/10.1016/j.compind.2019.08.004
9. X. D. Zhang, S. Y. Guo, J. Chen, D. F. Zhao, T. Yu, Simulation on human cooperation in production systems based on organization learning, *J. Industr. Eng. Eng. Manag.,* **27** (2013), 103–109. http://doi.org/10.3969/j.issn.1004-6062.2013.03.014
10. Y. Yadgarova, V. Taratukhin, An interoperable cloud environment of manufacturing control system, *Springer International Publishing,* **6** (2016), 3–12. http://doi.org/10.1007/978-3-319-30957-6_1
11. Y. K. Liu, X. Xu, L. Zhang, L. Wang, R. Y. Zhong, Workload-based multi-task scheduling in cloud manufacturing, *Robot Comput. Integr. Manuf.,* **45** (2016), 3–20. https://doi.org/10.1016/j.rcim.2016.09.008
12. Y. K. Liu, L. H. Wang, X. V. Wang, X. Xu, L. Zhang, Scheduling in cloud manufacturing: State-of-the-art and research challenges, *Int. J. Prod. Res.,* **57** (2019), 4854–4879. http://doi.org/10.1080/00207543.2018.1449978
13. B. V. Nouri, R. T. Moghaddam, M. Rohaninejad, A multi-objective scheduling model for a cloud manufacturing system with pricing, equity, and order rejection, *IFAC-PapersOnLine*, **52** (2019), 2177–2182. https://doi.org/10.1016/j.ifacol.2019.11.528
14. J. S. Ruiz, J. Mula, R. Poler, Smart manufacturing scheduling: A literature review, *J. Manuf. Syst.,* **61** (2021), 265–287. https://doi.org/10.1016/j.jmsy.2021.09.011
15. M. H. Yuan, Z. Zhou, X. X. Cai, C. Sun, W. B. Gu, Service composition model and method in cloud manufacturing, *Robot Comput. Integr. Manuf.,* **61** (2020), 101840. https://doi.org/10.1016/j.rcim.2019.101840

16. J. J. Zhou, X. F. Yao, Multi-population parallel self-adaptive differential artificial bee colony algorithm with application in large-scale service composition for cloud manufacturing, *Appl. Soft Comput.*, **56** (2017), 379–397. https://doi.org/10.1016/j.asoc.2017.03.017
17. T. R. Wang, P. Z. Zhang, J. Liu, M. M. Zhang, Many-objective cloud manufacturing service selection and scheduling with an evolutionary algorithm based on adaptive environment selection strategy, *Appl. Soft Comput.*, **112** (2021), 107737. https://doi.org/10.1016/j.asoc.2021.107737
18. H. Khojasteh, J. Misic, Task admission control policy in cloud server pools based on task arrival dynamics, *Wirel. Commun. Mob. Com.*, **16** (2016), 1363–1376. https://doi.org/10.1002/wcm.2689
19. Y. K. Liu, L. H. Wang, Y. Q. Wang, X. V. Wang, L. Zhang, Multi-agent-based scheduling in cloud manufacturing with dynamic task arrivals, *Procedia CIRP*, **72** (2018), 953–960. https://doi.org/10.1016/j.procir.2018.03.138.
20. S. Zhang, Y. B. Xu, W. Y. Zhang, Multitask-oriented manufacturing service composition in an uncertain environment using a hyper-heuristic algorithm, *J. Manuf. Syst.*, **60** (2021), 138–151. https://doi.org/10.1016/j.jmsy.2021.05.012
21. Y. J. Laili, S. S. Lin, D. Y. Tang, Multi-phase integrated scheduling of hybrid tasks in cloud manufacturing environment, *Robot Comput. Integr. Manuf.*, **61** (2020), 101850. https://doi.org/10.1016/j.rcim.2019.101850
22. C. Yang, T. Peng, S. L. Lan, W. M. Shen, L. H. Wang, Towards IoT-enabled dynamic service optimal selection in multiple manufacturing clouds, *J. Manuf. Syst.*, **56** (2020), 213–226. https://doi.org/10.1016/j.jmsy.2020.06.004
23. N. Xie, W. A. Tan, X. R. Zheng, L. Zhao, L. Huang, Y. Sun, An efficient two-phase approach for reliable collaboration-aware service composition in cloud manufacturing, *J. Ind. Inf. Integr.*, **23** (2021), 100211. https://doi.org/10.1016/j.jii.2021.100211
24. Y. X. Li, X. F. Yao, M. Liu, Cloud manufacturing service composition optimization based on reliability and credibility analysis, *Comput. Integr. Manuf. Syst.*, **27** (2021), 1780–1798. https://doi.org/10.1155/2019/7194258.
25. Y. F. Yang, B. Yang, S. L. Wang, T. G. Jin, S. Lin, An enhanced multi-objective grey wolf optimizer for service composition in cloud manufacturing, *Appl. Soft Comput.*, **2** (2019), 106003. https://doi.org/10.1016/j.asoc.2019.106003
26. Y. F. Yang, B. Yang, S. L. Wang, W. Liu, T. G. Jin, An improved grey wolf optimizer algorithm for energy-aware service composition in cloud manufacturing, *Int. J. Adv. Manuf. Tech.*, **105** (2019), 3079–3091.
27. D. Jones, C. Snider, A. Nassehi, J. Yon, B. Hicks, Characterising the digital twin: A systematic literature review, *Cirp J. Manuf. Sci. Tec.*, 29 (2020) 36–52. https://doi.org/10.1016/j.cirpj.2020.02.002
28. V. Vijayan, R. Harikrishnakumar, K. Krishnan, H. Cheraghi, S. Motavalli, Simulation-based decision framework for hybrid layout production systems under disruptions, *Proced. Manuf.*, **51** (2020), 1062–1068. https://doi.org/10.1016/j.promfg.2020.10.149
29. F. Psarommatis, A. Gharaei, D. Kiritsis, Identification of the critical reaction times for re-scheduling flexible job shops for different types of unexpected events, *Proced. CIRP*, **93** (2020), 903–908. https://doi.org/10.1016/j.procir.2020.03.038

30. J. P. Champati, B. Liang, Delay and cost optimization in computational offloading systems with unknown task processing times, *IEEE T Cloud Comput.*, **9** (2021), 1422–1438. https://doi.org/10.1109/TCC.2019.2924634
31. J. L. Risco-Martín, K. Henares, S. Mittal, L. F. Almendras, K. Olcoz, A unified cloud-enabled discrete event parallel and distributed simulation architecture, *Simul. Model Pract. Th.*, **118** (2022), 102539. https://doi.org/10.1016/j.simpat.2022.102539
32. C. Zhao, L. Zhang, Y. K. Liu, Z. Q. Zhang, G. J. Yang, B. H. Li, Agent-based simulation platform for cloud manufacturing, *Int. J. Model Simul. Sc.*, **8** (2017), 1742001. https://doi.org/10.1142/S1793962317420016
33. L. F. Zhou, L. Zhang, L. Ren, Simulation of production modes for cloud manufacturing enterprises, in *International Conference on Universal Village*, 2018, pp: 1–5. https://doi.org/10.1109/UV.2018.8642129
34. C. Zhao, X. Luo, L. Zhang, Modeling of service agents for simulation in cloud manufacturing, *Robot Comput. Integr. Manuf.*, **64** (2020), 101910. https://doi.org/10.1016/j.rcim.2019.101910
35. C. Zhao, L. Wang, X. Zhang, Service agent networks in cloud manufacturing: Modeling and evaluation based on set-pair analysis, *Robot Comput. Integr. Manuf.*, **65** (2020), 101970. https://doi.org/10.1016/j.rcim.2020.101970
36. Z. X. Zeng, Y. P. Yan, Multidimensional measure of cloud service composition flexibility, *Value Eng.*, **37** (2018), 281–283.
37. J. Yao, B. Xing, J. H. Wen, Survey on cloud manufacturing service composition, *Comput. Sci.*, **48** (2021), 245–255. https://doi.org/10.11896/jsjkx.200800173
38. M. K. Lim, W. Q. Xiong, Y. K. Wang, A three-tier programming model for service composition and optimal selection in cloud manufacturing, *Comput. Ind. Eng.*, **167** (2022), 108006. https://doi.org/10.1016/j.cie.2022.108006
39. Y. Liu, L. Zhang, Y. K. Liu, Y. J. Laili, W. C. Zhang, Model maturity-based model service composition in cloud environments, *Simul. Model. Pract. Th.*, **113** (2021), 102389. https://doi.org/10.1016/j.simpat.2021.102389
40. M. Eisa, M. Younas, K. Basu, I. Awan, Modeling and simulation of QoS-aware service selection in cloud computing, *Simul. Model. Pract. Th.*, **103** (2020), 102108. https://doi.org/10.1016/j.simpat.2020.102108

AIMS Press