IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 8, 2019, accepted December 22, 2019, date of publication December 24, 2019, date of current version February 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962093

A Novel Truthful and Fair Resource Bidding Mechanism for Cloud Manufacturing

ZHAOHUI LIU AND ZHONGJIE WANG, (Member, IEEE)
College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China
Corresponding author: Zhongjie Wang (wang_zhongjie@tongji.edu.cn)

This work was supported by the National Key Research and Development Program of China under Grant 2018YFB1702703.

ABSTRACT Cloud manufacturing is an emerging service-oriented and market-oriented networked manufacturing paradigm. Resource scheduling of cloud manufacturing is different from those of other manufacturing paradigms due to some distinctive characteristics, such as high concurrency of multi-service and multi-task, high heterogeneity and high dynamics of resources, incomplete information of cloud services and user-centric services. These characteristics make the resource scheduling problem of cloud manufacturing more complicated. Efficient resource scheduling and economic benefit are the major concerns of users in cloud manufacturing environment. A novel market-based continuous bidding mechanism is proposed in this paper by applying game theory. In this mechanism, the overall benefit of both the cloud service provider and the cloud service demander is considered as the optimization target. The constraints are the task delivery time and budget. For decomposable tasks, the constraints for each subtask are also considered in resource bidding. Service demanders can bid continuously until the corresponding service is obtained or the bidding constraint is violated. Experiments demonstrate the proposed resource bidding mechanism is efficient in cloud service scheduling and can ensure the cloud manufacturing market is truthful and fair.

INDEX TERMS Cloud manufacturing, resource scheduling, bidding mechanism, decomposable task, workflow.

I. INTRODUCTION

In recent years, with the development of information technology and network technology, the manufacturing paradigm has experienced fundamental changes, from virtual manufacturing [1], agile manufacturing [2] and grid manufacturing [3], to the current cloud manufacturing [4], [5]. Cloud manufacturing is an advanced service-oriented and market-oriented manufacturing paradigm. Based on cloud computing [6] and Internet of Things(IoT) [7], resources are treated as services, cloud manufacturing provides green, intelligent, collaborative, high-efficient and on-demand manufacturing services by establishing manufacturing resource pool for users. The cloud manufacturing services will be matched with the tasks submitted to the cloud manufacturing platform based on some resource scheduling algorithms. These tasks could be either atomic or decomposable. When they are decomposed, the subtasks constitute the workflow of the original task according to the processing dependencies. The problem of task decomposable resource scheduling is one of the key issues for cloud manufacturing.

The associate editor coordinating the review of this manuscript and approving it for publication was Asad Waqar Malik.

Different from the other manufacturing paradigms, cloud manufacturing has the following characteristics. First, the manufacturing resources and capabilities, including production, design, simulation, manpower and knowledge, etc, are highly heterogeneous and dynamic. These resources and capabilities are free to get access into and drop out of the cloud manufacturing resource pool at any time [8], [9]. The manufacturing tasks are at different levels, being either atomic or decomposable, and the number of decomposable tasks is increasing [10], [11]. The manufacturing services and tasks are mutually selected because cloud manufacturing is market-oriented [12], [13]. The manufacturing tasks are highly concurrent and diverse [14]–[16]. User-centric service is one of the key characteristics of cloud manufacturing, users would like to consider their preferences and personalized needs when choosing services [17]–[19]. Therefore, the resource scheduling faces more challenges in cloud manufacturing.

First, the resource scheduling of cloud manufacturing has to satisfy the constraint of the task workflow [20]. A task workflow consists of more than one subtask node. Each subtask will not start being processed until those subtasks before it are finished. The constraints of subtasks can be evaluated and used to optimal resource scheduling [21]–[23]. Liu [24] discusses the resource scheduling of instance-intensive workflows in cloud computing platform. Scheduling algorithm in [24] takes the execution cost and the execution time as two key considerations, considers sharing, conflicting and competition of services caused by multiple concurrent instances, allocates sub-deadlines to subtasks of each instance workflow, calculates the estimated execution time and cost for each service, and then allocates the services to the tasks. This algorithm can achieve a lower cost than others while meeting the user designated deadline or reduce the mean execution time with lower cost. After a manufacturing task is transformed into a workflow, some literatures have discussed the effects of subtask on resource scheduling in cloud manufacturing [25]–[28]. Zhou *et al.* [28] proposes an improved genetic algorithm for optimal task scheduling solutions with diverse tasks. The algorithm considers the dependence of subtask and much more service attributes. Zhou indicates quality defects of front-end services have a larger impact on quality of the whole manufacturing tasks than back-end services. These literatures only consider the process constraint of subtask and coarse-grain constraints of manufacturing task. However, the sub-budget and sub-delivery time also do affect resource scheduling in cloud manufacturing. In addition, many scheduling algorithms based on heuristic algorithms are passive, participants are capable of violating a passive scheduling solution for pursuing the maximization of individual interest.

Secondly, the information is not completely transparent for the participants to the transaction. In order to make a benefit balance among the participants, negative behaviors, such as cheating, have to be prohibited. There are some literatures guaranteeing quality of service based on service level agreements, if a provider cannot satisfy its commitment, it should pay penalties as compensation [29], [30]. While individual benefits may be inconsistent when different demanders compete for the same resource or providers undertake the same task, some researchers employ game theory [31] to solve this problem. Fard *et al.* [32] designed a truthful mechanism considering both the cost and the makespan for workflow scheduling. However, this mechanism does not investigate the demander's bidding strategies and is not suitable for multiple heterogeneous tasks competing for the resources simultaneously. Nezarat and Dastghaibifard [33] designed a bidding mechanism from the perspective of cloud service demanders. In this mechanism, the extensible utility function of service providers is used, the delivery time and budget of service demanders are taken as the constraints, and Bayesian forecast theory is employed to make the price strategy. As a result, both the demander and the provider achieve the greatest benefits. However, this mechanism assumed that the subtasks are parallel and mutually independent.

In order to allocate resources efficiently in multi-service and multi-task application with incompletely transparent information, a novel truthful and fair resource bidding mechanism for cloud manufacturing is proposed in this paper. In this mechanism, the service providers are required to offer a price honestly and can obtain economic benefit in incompletely transparent information environment. When the service demanders make a price, they could consider many factors, such as delivery time, budget and user preferences. Meanwhile, fair competition for the resources can be guaranteed, and the resource scheduling is Nash equilibrium.

The remainder of this paper is organized as follows. Section 2 gives the problem description and modeling. The bidding mechanism design is discussed in detail in section 3. Section 4 gives the pseudo code of this bidding mechanism. Simulation and application are shown in section 5. Section 6 is the conclusion.

II. PROBLEM DESCRIPTION AND MODELING

A. PROBLEM DESCRIPTION

In cloud manufacturing environment, resources are treated as services, so these two words will be used alternatively in the rest of this paper. Because the resources are heterogeneous, the resource scheduling for multi-task may be concurrent. Each service demander may have to compete with others in addition to meeting its own process constraints when it applies for resources. Furthermore, the information is not completely transparent, both the service demanders and the service providers offer their prices back to back. For service demanders, they expect to acquire high quality resources with as low price as possible. While for service providers, they expect as high price as possible. How to make a benefit balance between demanders and providers is the main problem to be solved in cloud manufacturing environment. In addition, high dynamics and uncertainty are ubiquitous in cloud manufacturing. Manufacturing resources are free to be released into the resource pool or retrieved from it. Manufacturing tasks may arrive randomly. These characteristics impose great challenges on cloud manufacturing resource scheduling.

B. SCHEDULING MODEL

For simplicity, there are some assumptions in the scheduling model. One service is prohibited to be allocated to new tasks until it is released, and one subtask can not start being processed until the subtasks before it are finished. The mapping relationship between demanders and tasks is one-to-one relationship, so do providers and services. The mapping relationship between subtask types and service types is also defined as one-to-one relationship.

The set of service demanders is denoted by RD and the number of service demanders is denoted by M. The set of tasks is $Task = \{task^1, task^2, \cdots, task^M\}$. $task^i$ denotes the task submitted by demander i, $i \leq M$. $task^i$

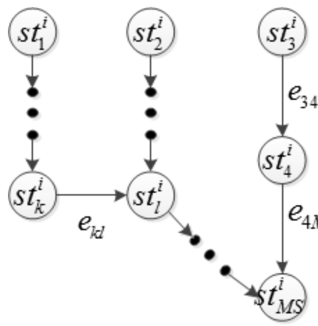


FIGURE 1. Workflow of $task^i$.

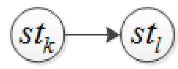


FIGURE 2. Serial subtasks.

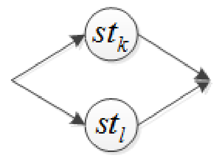


FIGURE 3. Parallel subtasks.

can be described as an array with five elements, $task^i = (task^i\_id, task^i\_name, task^i\_type, task^i\_delivery, task^i\_budget)$, the means of each element are the index, name, type, delivery time, and budget of task i respectively. $task^i\_delivery$ and $task^i\_budget$ are the constraints of task and cannot be violated. The set of service providers is denoted by RP and the number of service providers is denoted by N. The set of services is $Service = \{Service^1, Service^2, \cdots, Service^N\}$. $Service^j$ denotes the service provided by provider j, $j \leq N$, $Service^j$ can be described as an array with five elements, $Service^j = \{Service^j\_id, Service^j\_name, Service^j\_ST, Service^j\_ET, Service^j\_cost\}$, the means of each element are the index, name, supporting task type, execution time, and price of service j respectively.

1) TASK WORKFLOW

The directed acyclic graph (DAG) is employed to describe the workflow structure of tasks, as shown in Fig. 1.

In Fig. 1, $task^i$ has MS subtasks, st is the short of *subtask*. Each vertex denotes one subtask, let V denote the vertex set of DAG. For task i, the subtask set can be described as,

$$V^i = (subtask_1^i, subtask_2^i, \cdots, subtask_k^i, \cdots, subtask_{MS}^i).$$

$subtask_k^i$ can be described as an array with five elements,

$$st_k^i = (st_k^i\_id, st_k^i\_name, st_k^i\_type, st_k^i\_delivery, st_k^i\_budget)$$

the means of each element are the index, name, type, delivery time, and budget of $subtask_k^i$ respectively.

The edge between two nodes in DAG denotes the relationship between the two corresponding subtasks, let E denote the edge set of DAG. For task i,

$$E^i = \begin{bmatrix} e_{11}^i & \cdots & e_{1l}^i & \cdots & e_{1MS}^i \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ e_{k1}^i & \cdots & e_{kl}^i & \cdots & e_{kMS}^i \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ e_{MS1}^i & \cdots & e_{MSl}^i & \cdots & e_{MSMS}^i \end{bmatrix}.$$

e_{kl}^i denotes the interrelationship between subtask k and subtask l of task i, let $succ^i(k)$ and $pre^i(k)$ denote the successors and the predecessors of subtask k.

$$e_{kl}^i = \begin{cases} 1, & \text{if } subtask_l^i \in succ_k^i \\ 0, & \text{otherwise} \end{cases}$$

if $pre^i(k) = \emptyset$, it means $subtask_k^i$ is the starting node of DAG^i; if $succ^i(k) = \emptyset$, it means $subtask_k^i$ is the end node of DAG^i. Usually, there is only one end node of each task.

In DAG, the structure among subtasks can be either serial or parallel. For example, $subtask_k$ and $subtask_l$ are serial in Fig. 2, and parallel in Fig. 3.

Calculation of the total execution time and cost for these two structures is quite different, as shown in Table 1. ET_k denotes the execution time of subtask k, $cost_k$ is the execution cost of subtask k.

TABLE 1. Calculation of execution time and cost.

subtask structure	execution time	cost
serial	$ET_k + ET_l$	$cost_k + cost_l$
parallel	$\max\{ET_k, ET_l\}$	$cost_k + cost_l$

2) USER PREFERENCE

In cloud manufacturing environment, besides the process constraint is taken into account, the user preference of service demanders and services providers also have a great influence on the bidding price. Let matrix $P \in R^{M \times N}$ represents the relationship between demanders and providers. M and N are the number of demanders and providers.

$$P = \begin{bmatrix} p^{11} & \cdots & p^{1j} & \cdots & p^{1N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ p^{i1} & \cdots & p^{ij} & \cdots & p^{iN} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ p^{M1} & \cdots & p^{Mj} & \cdots & p^{MN} \end{bmatrix}$$

p^{ij} measures the intimacy between service demander i and service provider j. Greater value of p^{ij} indicates closer partnership between demander i and provider j, and demander i has more preference to select service j. When bidding for services, demander i also has more motivation to bid a higher price to win service j.

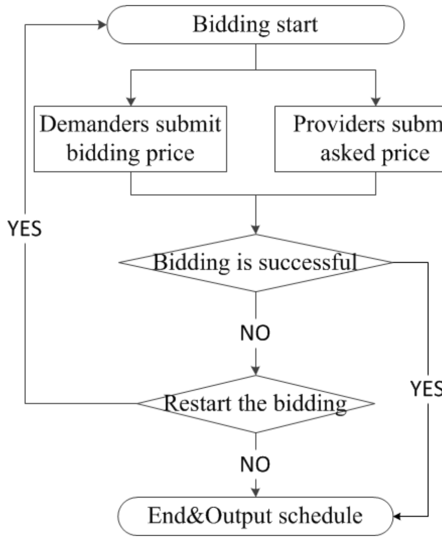


FIGURE 4. Diagram of cloud manufacturing resource bidding mechanism.

3) BENEFIT

The diagram of cloud manufacturing resource bidding mechanism is shown in Fig. 4. When the bidding process starts, both the demanders and the providers submit their prices, called bidding price and asked price. The bidding process will be over until all the tasks find their services or the bidding completion condition is met.

Let $Schedule = \{Sch_1, Sch_2, \cdots, Sch_a, \cdots, Sch_d\}$ denote the resource scheduling scheme. For example, $Sch_a = \{subtask_k^i, Service^j\}$ means $Service^j$ is allocated to $subtask_k^i$, the benefits of both sides can be evaluated by formula (1).

$$\begin{aligned} U_d^i(k) &= bid_k^i - pay(a) \\ U_p^j &= pay(a) - cost^{jr} \end{aligned} \tag{1}$$

$U_d^i(k)$ is the benefit of demander i for subtask k, U_p^j is the benefit of provider j in this transaction, bid_k^i is the bidding price for $subtask_k^i$, $pay(a)$ is the value of transaction a between demander i and provider j, $cost^{jr}$ is the real cost of provider j.

In conclusion, the scheduling model is as follows.

Optimization objective:

$$Max\left\{U_d^i(k), U_p^j\right\}$$

Constraints:

$$\begin{aligned} &\sum_{k=1}^{MS} bid_k^i \leq task^i\_budget, \\ &Delivery^i \leq task^i\_delivery, \\ &pre^i(k) \subseteq V^i - Res^i(k). \end{aligned}$$

where, $Delivery^i$ is the final delivery time of task i, $Res^i(k)$ is the set of the subtasks that have not been matched with services besides $subtask_k^i$. The last constraint means $subtask_k^i$ will not start bidding for service until those subtasks before it are finished.

III. BIDDING MECHANISM

A. PRIORITY OF SUBTASKS

After one task is decomposed, the subtasks are interdependent, The first step is to determine the priority of each subtask in the workflow before selecting the cloud services. In order to ensure the task to be properly completed, the cloud services have to be selected based on the priorities.

For $subtask_k$, the priority can be calculated by formula (2),

$$rank_k = \begin{cases} load_k + \max\limits_{l \in succ(k)}\{load_l\}, & if\ succ(k) \neq \emptyset \\ load_k, & if\ succ(k) = \emptyset \end{cases} \tag{2}$$

The value of $rank_k$ represents the distance between $subtask_k$ and the end node subtask. The greater $rank_k$ is, the higher the priority of $subtask_k$ is. $load_k$ denotes the workload of $subtask_k$.

B. BIDDING PRICE OF SUBTASK

When the original task is decomposed, each subtask has to participate in resource bidding. Usually, the demander only gives the total budget and the final due date for original task. When bidding for a resource, the factors to be considered include budget, delivery time and individual preference as well. For $subtask_k^i$, the bidding price is given in formula (3).

$$bid_k^{i,count} = \rho_k^i * [e^{\Delta} * V_{\max}^{i,count}(k) + (1 - e^{\Delta})V_{\min}^{i,count}(k)] \tag{3}$$

There are three main parts in the bidding algorithm. $V_{max}^{i,count}$ and $V_{min}^{i,count}$ reflect the influence of budget of $subtask_k^i$ on the bidding price. Δ reflects the influence of delivery time of $subtask_k^i$ on the bidding price. ρ_k^i reflects the influence of user preference on the bidding price. Using this bidding strategy demanders can make some economic benefit except for process requirement being satisfied even if they just know their own information.

1) CALCULATION OF $V_{max}^{i,count}$ and $V_{min}^{i,count}$

If a demander intends to obtain certain services, the bidding price should not be less than the lowest price of these services, denoted by $V_{min}^{i,count}(k)$, $count$ indicates the times of demander bidding for $subtask_k^i$. The upper bound, $V_{max}^{i,count}(k)$, can be calculated by formula (4).

$$\begin{aligned} V_{max}^{i,count}(k) = OverallBudget^i - PreCost^i(k) \\ - min \sum_{\bar{k} \in Res^i(k)} cost_{\bar{k}}^i \end{aligned} \tag{4}$$

$OverallBudget^i$ is the total budget of demander i. $PreCost_k^i$ is the total cost of those subtasks that have been matched successfully with the services at the time of bidding $subtask_k^i$. $Res^i(k)$ is the set of the subtasks that have not been matched with services besides $subtask_k^i$.

Inference 1: If any subtask exceeds the upper bound of price limit, the total cost of the entire task workflow must exceed the total budget.

Inference 1 can be verified through formula (4).

2) CALCULATION OF Δ

In manufacturing industry, closer delivery time usually leads to more urgent demand, thereby high price is promoted when bidding for high quality service as soon as possible.

$$SubDelivery_k^{i,count} = Delivery^i - ST^{count} - PreTime^i(k) - min \sum_{\bar{k} \in ToE^i(k)} ET_{\bar{k}}^i \tag{5}$$

$SubDelivery_k^{i,count}$ is the delivery time constraint of $subtask_k^i$. $Delivery^i$ is the final delivery time of task i, ST^{count} is the moment of the $count$th bidding. $PreTime_k^i$ is the total processing time required for the subtasks of $task^i$ that have been assigned services. $ToE^i(k)$ is the set of all the subtasks after $subtask_k^i$ along the longest path to the end subtask. The initial moment is ST^0. ST^{count} will be updated after each bidding, the updating method is as formula (6).

$$ST^{count} = count \times T + ST^0 \tag{6}$$

T is the average period of each bidding.

Before bidding for $task_k^i$, there is one parameter that shouldn't be neglected, it is the buyer's maximum preparation time, denoted by $Readytime_k^{i,count}$, calculated by formula (7).

$$Readytime_k^{i,count} = SubDelivery_k^{i,count} - minET_k^i \tag{7}$$

The $minET_k^i$ represents the minimum processing time required by the service candidates supporting $subtask_k^i$.

Inference 2: If any subtask violates its sub-delivery limit, the entire task workflow must violates the total delivery time.

Δ is calculated by formula (8).

$$\Delta = -\frac{Readytime_k^{i,count}}{\alpha T} \tag{8}$$

α is an adjustable constant.

3) CALCULATION OF ρ_k^i

The elements of matrix P could reflect the intimacy between service demanders and service providers. When a demander faces more than one provider, the intimacy will be the average value of all the candidate providers in P,

$$\bar{p}_k^i = \frac{\sum_{CS} p^{ij}}{Amount_k^i(CS)} \tag{9}$$

$\bar{p}_k^i$ is the average value of the candidate services supporting $subtask_k^i$. CS is the set of available candidate services. $Amount_k^i(CS)$ is the number of all the candidate services supporting $subtask_k^i$. Sigmoid function can perfectly approximate the user sensitivity to intimacy [34]. Factor ρ is defined as formula (10).

$$\rho_k^i = \begin{cases} 1 \bar{p}_k^i = p_{\max} \\ \dfrac{1}{1 + e^{-\beta(\bar{p}_k^i - \frac{p_{\max} - p_{\min}}{2})}} \bar{p}_k^i \in (p_{\min}, p_{\max}) \\ 0 \bar{p}_k^i = p_{\min} \end{cases} \tag{10}$$

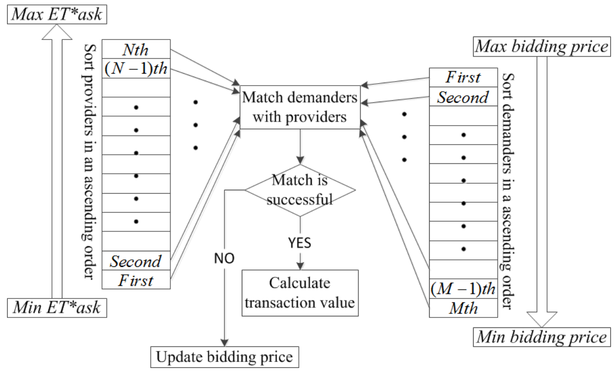


FIGURE 5. Matching process.

p_{max} and p_{min} are the maximum and the minimum value of the relationship degree. β is the zoom factor, which can adjust the curve shape based on the interval of relationship degree.

C. PRICE STRATEGY OF PROVIDER

Let ask^j and ET^j denote the asked price and the makespan of service provider j. Service providers only know the processing time and real cost of their own services, and are unaware of these information of other services. In the case of opaque information, in order to ensure their own benefits, the rational strategy is to ask for price honestly. So, the truthful strategies of providers are Nash equilibrium in the process of bidding mechanism. When a service provider submits the asked price, the makespan of the corresponding task must be submitted as well.

D. TRANSACTION VALUE

At the end of each round of bidding process, the service demanders are sorted in a descending order according to their bidding prices, and the service providers are sorted in an ascending according to $ET * ask$. The appropriate services are allocated to demanders in turn, as shown in Fig. 5. There are three criterions to judge whether the match is successful.

- The resource must support the task submitted by demanders.
- The resource must be available.
- The bidding price must be greater than the asked price.

If the match is successful, the demander and the first provider are winners.

In a general double auction, the transaction value between demander i and provider j refers to the average of the bidding price and the asked price, as shown in formula (11).

$$pay(i, j) = \frac{1}{2} \times (bid_k^{i,count} + ask^j) \tag{11}$$

However, this pricing method does not guarantee the asked price is truthful. In order to make the providers to ask for a truthful price, the Second Price Sealed Auction is employed to construct the transaction value in this paper, as shown in formula (12).

$$pay(i,j) = \begin{cases} pay' & pay' \in (ask^j, bid^i) \& ET^{jr} \leq ET^j \\ bid_k^{i,count} & pay' \notin (ask^j, bid^i) \& ET^{jr} \leq ET^j \\ pay^* & ET^{jr} > ET^j \end{cases} \tag{12}$$

where,

$$pay' = \frac{1}{2} \times (bid_k^{i,count} + \frac{ask^{j+1} * ET^{j+1}}{ET^j}),$$

the service provider $(j+1)$ is the first provider after provider j in the sorted RP, ET^{jr} is the real execution time of service j.

$$pay^* = 0.5 * \min_{j \in [1,\ldots,N]} (cost^{jr}).$$

When $ET^{jr} \leq ET^j$ and $pay' \in (ask^j, bid^i)$, $pay(i,j)$ is pay'.

When $ET^{jr} \leq ET^j$ and $pay' \notin (ask^j, bid^i)$, it is known that,

$$bid_k^{i,count} \geq ask^j, \left(ask^{j+1} * ET^{j+1}\right)/ET^j \geq ask^j. \tag{13}$$

However, it is unknown that the relationship between $\left(ask^{j+1} * ET^{j+1}\right)/ET^j$ and $bid_k^{i,count}$. To protect the economic benefit of demanders and don't harm that of providers at the same time, when $bid_k^{i,count} \leq (ask^{j+1} * ET^{j+1})/ET^j$, define $pay(i,j) = bid_k^{i,count}$.

When $ET^{jr} \leq ET^j$, provider j provides a false information and should be punished, the mechanism in this paper inflicts punishment on the corresponding service provider j. Let $pay(i,j) = pay^*$, pay^* is defined as a penalty function, which is half-value of the minimum service price taking part in bidding process.

After one task is finished, the real execution time could be known, then the transaction value would be calculated based on it. If the cloud service provider fails to finish on time, the penalty function will be enabled.

According to formula (12), it is known that,

$$cost^{jr} < pay(i,j) \leq bid_k^{i,count},$$

The demander will pay less than, with the minimum probability equal to, of being its own bidding price. In this mechanism, both the demanders and the providers are more effective, and do not have to worry detriment of their own interests caused by opaque market information. The mechanism can make the demanders and the providers avoid benefit loss in cloud manufacturing, therefore this mechanism design has the price protection function.

Theorem 1: Let s_* represents the honest strategy profile of the participants in bidding mechanism, if the participants are rational, s_* is Nash equilibrium.

Proof: In the opaque information environment, the provider's economic benefit is calculated in formula (14).

$$U_p^j = pay(i,j) - cost^{jr} \tag{14}$$

If $(ET^j, ask^j) \neq (ET^{jr}, cost^{jr})$, there will be four cases.
leftmargin=2.75em

case1 $ET^j > ET^{jr}, ask^j > cost^{jr}$
This strategy leads to a lower rank and a less winning probability. Even if the provider wins in cloud service competition, because

$$\left(ask^{j+1} * ET^{j+1}\right)/ET^j < \left(ask^{j+1} * ET^{j+1}\right)/ET^{jr}.$$

This strategy will reduce the revenue of service providers.

case2 $ET^j < ET^{jr}, ask^j > cost^{jr}$
even if wins in auction, because of the penalty, the ultimate benefit is negative.

$$\begin{aligned} U_p^j &= pay(i,j) - cost^{jr} \\ &< min_{\tilde{j} \in [1,\ldots,N]}(cost^{\tilde{j}r}) - cost^{jr} \\ &\leq 0 \end{aligned}$$

case3 $ET^j > ET^{jr}, ask^j < cost^{jr}$
This strategy not only reduces the income of provider, but also reduces the chances of winning.

case4 $ET^j < ET^{jr}, ask^j < cost^{jr}$
The provider reduces the asked price although it can increase the probability of winning, if $pay(i,j) < cost^{jr}$, $U_p^j < 0$. If $ET^j = ET^{jr}$, $pay' \in (ask^j, bid^i)$, the asked price of provider j is $cost^{jr}$. The benefit of service provider j can be shown as formula (15).

$$\begin{aligned} U_p^j &= pay(i,j) - cost^{jr} \\ &= \frac{1}{2} \times (bid_k^i + \frac{ask^{j+1} * ET^{j+1}}{ET^j}) - cost^{jr} \end{aligned} \tag{15}$$

The asked price of provider j is $cost^j$, let $(ET'^{j+1}, cost'^{j+1})$ denote the performance parameters of service $(j+1)$ when $ask^j < cost^{jr}$, and $(ET^{j+1}, cost^{j+1})$ denoting the parameters when $ask^j = cost^{jr}$. The benefit of provider j can be given in formula (16).

$$\begin{aligned} U_p'^j &= pay(i,j) - cost^{jr} \\ &= \frac{1}{2} \times (bid_k^i + \frac{ask'^{j+1} * ET'^{j+1}}{ET^j}) - cost^{jr} \end{aligned} \tag{16}$$

And because

$$\begin{aligned} ET^{jr} * cost^j &\leq ET'^{j+1} * cost'^{j+1} \\ ET^{jr} * cost^{jr} &\leq ET^{j+1} * cost^{j+1} \\ ET^j * cost^j &< ET^{jr} * cost^{jr} \\ ET'^{j+1} * cost'^{j+1} &\leq ET^{j+1} * cost^{j+1} \end{aligned}$$

So $U_p'^j < U_p^j$, that is, the benefit of provider j will be reduced. It is also irrational to take the strategy of reducing benefits.

In conclusion, the truthful strategy profile for all service providers is Nash equilibrium. □

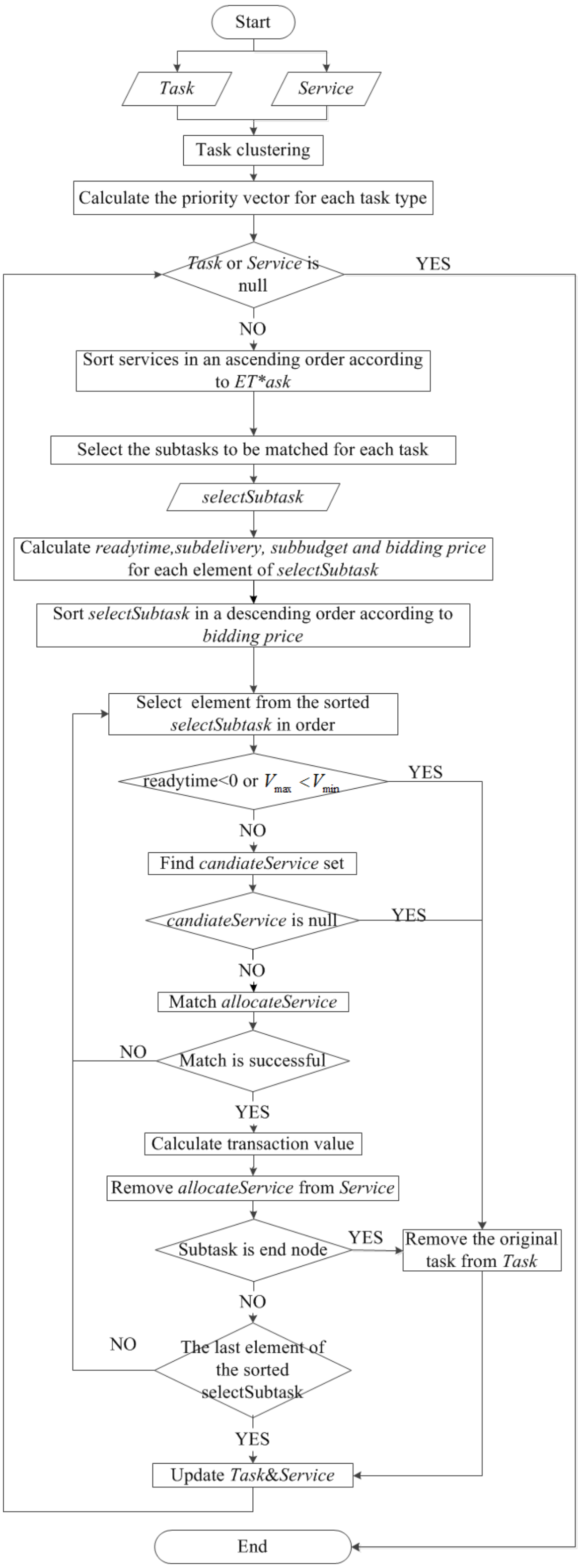


FIGURE 6. Flow chart of the proposed bidding mechanism.

IV. PSEUDO CODE OF THE ALGORITHM

Let M denote the number of tasks and N is the number of services, and all the tasks can be divided into X classes with different DAG. The process of the proposed resource bidding mechanism is shown in Fig. 6.

step 1 Calculate the priority vector for each task type.

step 2 If the task set and service set are not null, go to next step, otherwise end program.

step 3 Sort the services according to the product of *ET* and *cost* and generate a new service set.

step 4 Select the subtasks to be matched and generate *selectSubtask*.

step 5 Calculate *readytime*, *sub-delivery time*, *sub-budget* and *bidding price* for each selected subtask.

step 6 Sort *selectSubtask* in a descending order according to the bidding prices and generate a new subtask set.

step 7 Match services for each subtask. If a subtask violates its sub-constraints or there are not services supporting this subtask, the corresponding original task will fail bidding and be removed from *Task*.

step 8 If the match is successful, calculate the transaction value and remove the corresponding service from *Service*, otherwise go back to step 7.

step 9 If the corresponding original task hasn't been finished, go back to step 7, otherwise remove the corresponding original task from *Task*.

step 10 After each subtask of *selectSubtask* is matched, update *Task* and *Service*. In this step, the tasks, which have obtained enough services or failed in bidding, are removed, the occupied and faulty services are also removed. New arriving tasks and access services will join in the next round of bidding. And then, go back to step 2.

The pseudo code for the proposed resource bidding mechanism is shown in Algorithm 1.

V. EXPERIMENTS

A. SIMULATION

The proposed cloud manufacturing resource bidding mechanism is investigated and compared with the earliest due date(EDD) algorithm and genetic algorithm(GA). In the simulation experiment, $X = 3$, $M = 300$, $N = 3400$. As shown in Fig. 7, three different task types are denoted by *type*1, *type*2, *type*3. Each task type has 100 users and each subtask has 200 services, the delivery time of the tasks of these three task types are randomly generated in three intervals, [70, 150], [90, 200] and [65, 160], the unit is day. The budgets are also randomly generated in [1.8, 2.5], [3.3, 4.0] and [1.7, 3.0], the unit is million dollars.

$$\begin{aligned} rank(type1) &= [st_1^1, st_2^1, st_3^1, st_4^1, st_5^1] \\ rank(type2) &= \left[st_1^2, st_2^2, st_3^2, st_4^2, st_5^2, st_6^2, st_7^2\right] \\ rank(type3) &= [st_1^3, st_2^3, st_3^3, st_4^3, st_5^3] \end{aligned} \tag{17}$$

1) BIDDING RESULT

In the bidding mechanism, the winners of demanders can get the cloud manufacturing services. For example, demander

Algorithm 1 Algorithm of the Proposed Bidding Mechanism

Input: the set of manufacturing tasks, *Task*; the set of cloud services, *Service*; the matrix of relationship degree, *P*; the set of DAGs' models, *DAG*; the initialization parameter ST^0, T, α, β

Output: the result of workflow scheduling, *Schedule*;

```
for each element of DAG do
    for each subtask of manufacturing task do
        Calculate the priority
    end for
    Get priority vector of each element of DAG
end for
while Task ≠ ∅&Service ≠ ∅ do
    for each element of Service do
        Submit ET&ask
    end for
    Sort descending Service according to ET * ask
    for each element of Task do
        Get the priority vector of task
        Select subtasks whose priorities are greater and predecessors are finished, generate selectSubtask.
        Calculate SubDelivery, SubBudget, readytime, bidding price
    end for
    Sort selectSubtask in an ascending order according to bidding price
    for each element of bidder do
        if Readytime < 0 or V_max < V_min then
            Remove task from Task
        else
            Find candiateService set
            if candiateService = ∅ then
                Remove task from Task
            else
                Match allocateService
                Calculate pay
                Remove allocateService from Service
                if the subtask is end node of task then
                    Remove task from Task
                end if
            end if
        end if
    end for
    Update Task, Service
end while
```

3 submits a task of *type*3, $task^3\_delivery = 90$ days and $task^3\_budget = 2.7$ million dollars. The due date and budget for each subtask can be shown in Fig. 8.

Only 61 days and 1.99 million dollars are required for completing this task, the demander's requirement is fully satisfied. The cloud services allocated to [$subtask_1^3$, $subtask_2^3$, $subtask_3^3$, $subtask_4^3$, $subtask_5^3$] are ([d3t1s0016, d3t2s0001, d3t3s0158, d3t4s0064, d3t5s0192]). The corresponding

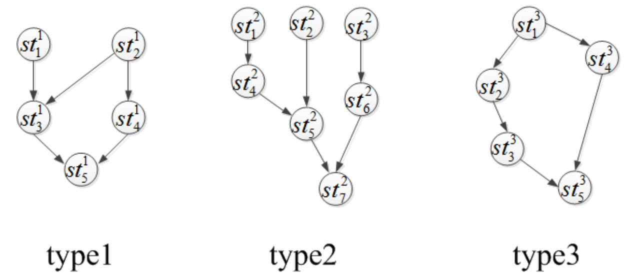


FIGURE 7. DAG model of tasks.

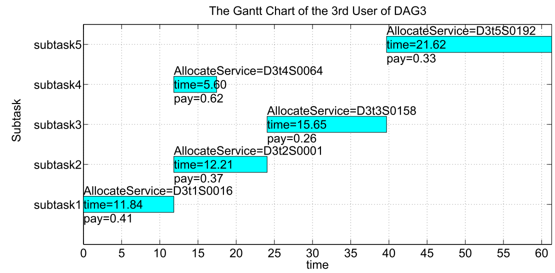


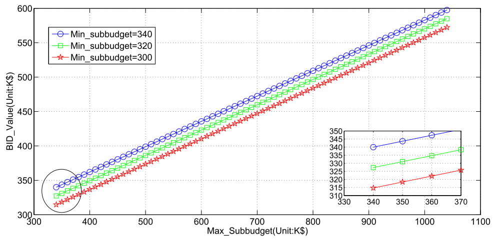
FIGURE 8. Gantt of $task^3$.

transaction values are [0.41, 0.37, 0.26, 0.62, 0.33], and the bidding prices are [0.51, 0.42, 0.3, 0.63, 0.34], the asked prices are [0.31, 0.3, 0.22, 0.61, 0.32]. It is clear that the transaction values are lower than the bidding prices and higher than the asked prices. The participants will make more economic benefit except for the process requirement being satisfied when they just know their own information.

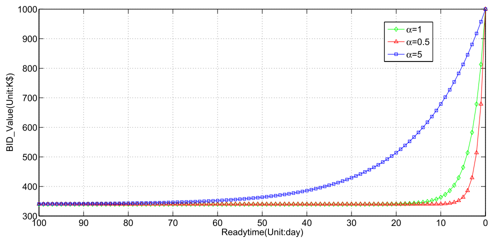
2) INFLUENCE OF FACTORS ON BIDDING PRICE

Bidding price is related to the constraints of the manufacturing task and individual subjective preference. Take as an example to analyze the influences of budget, delivery time, and preference on the bidding price. The result is shown in Fig.9.

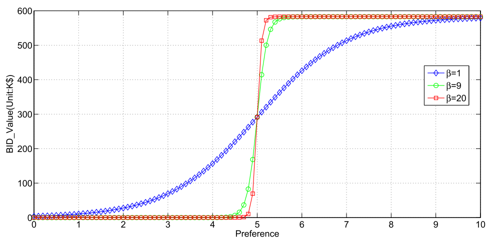
Fig. 9(a)-9(c) show the influence of a single factor on bidding price. The bidding price is directly proportional to the budget, while the bidding price will decrease with the lower bound of the budget down. Fig. 9 fully illustrates that the calculation of bidding price not only can help the demander win the auction with a budget constraint as much as possible, but also be closely correlated with the service market. The bidding price and ready time are in an exponential relationship. As the maximum ready time is reduced, the resource demand becomes more urgent, and the bidding price increases further. Bidding price and preference are nonlinear. The subjective factors, and, will vary with different demanders. They could express the sensitivity of demanders to the delivery time and preference. Fig. 9(d) depicts a four-dimensional diagram, which shows the comprehensive influence of various factors on bidding price, indicated by colors. The bidding price is within a reasonable range. The bidding price will reach the limit value only if all factors reach the limit.



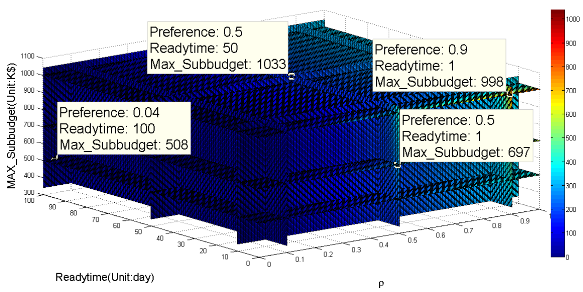
(a) The influence of the budget



(b) The influence of the ready time



(c) The influence of the preference



(d) Comprehensive influence of factors

FIGURE 9. **Influence of factors on bidding price.**

3) RESOURCE SCHEDULING EFFICIENCY

The proposed resource scheduling method, shorted NBM, is compared with EDD algorithm and GA. For EDD, the priorities of tasks are determined only by the delivery time. The shorter the delivery time is, the higher the priority is, EDD doesn't consider the other factors. In the experiment of GA, the crossover rate and mutation rate are 0.8 and 0.01. The optimization objective is efficiency, delivery time, budget and preference are considered to optimize scheduling solution. The scheduling efficiency is shown in Fig. 10.

For the three types of tasks, *type*1, *type*2, *type*3, the number of winners by NBM are 79, 75 and 89, only 41, 44 and 74 by EDD, and 70, 69 and 66 by GA. It is clear that the success percentage of NBM is higher than that of EDD and GA.

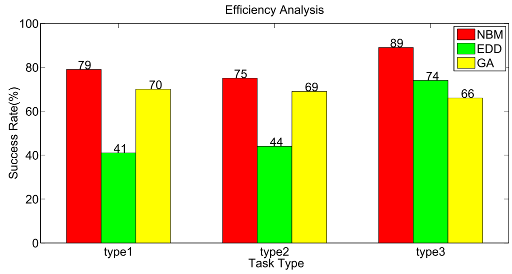


FIGURE 10. **Success percentage of NBM&EDD&GA.**

4) FAIRNESS

In addition, only fair and impartial cloud services trading market can enhance the enthusiasm of the participants to ensure sustainable and healthy market development. The proposed resource scheduling mechanism ensures that the service competition is fair.

Fig. 11 shows the delivery time and budget for all the tasks of *type*3. Fig. 12 shows the scheduling results of NBM, EDD and GA. The green ones represent the demanders succeeded in obtaining services, and the red ones indicate the losers. Fig.11 and Fig.12 show that the demanders with short delivery time and low financial budget will have a higher probability of failure in these two algorithm. However, the demanders with high budget will have different result, though they have shorter delivery time as well. For example, in EDD,demander 59, 61 do not get the cloud services they need, and demander 60 obtains the required services successfully with lower budget. The reason is that the delivery time of demander 60 is relatively shorter compared with demander 59 and 61, and the services meeting the requirements of demander 59 and 61 have been taken up by demander 60, even if demander 60 has lower financial budget. Obviously, this scenario is impossible in real market transactions, any rational demander will try desperately to obtain the services for their economic benefit, it is also a more realistic and effective way for demanders to raise the bidding price to improve its priority of accessing to services, there is no reason for any rational service provider to give up trading that increases its economic benefit. So EDD is unfair to the users with sufficient budget and urgent need of services, and the overall social benefit will be decreased. In NBM, demander 59 and 60 will win in the fierce competition by improving their bidding prices based on their own sufficient budget, meanwhile the economic benefit of providers will be increased. The services will be utilized with high efficiency.

5) INITIATIVE

In cloud manufacturing environment, service demanders would like to select services according to their preference, instead of passively accept. The proposed algorithm can solve this problem well. The details of 7 tasks of *type*3 are given in Table 2.

The 7 tasks will compete for the same services. The second column is the rank value by delivery time. The shorter

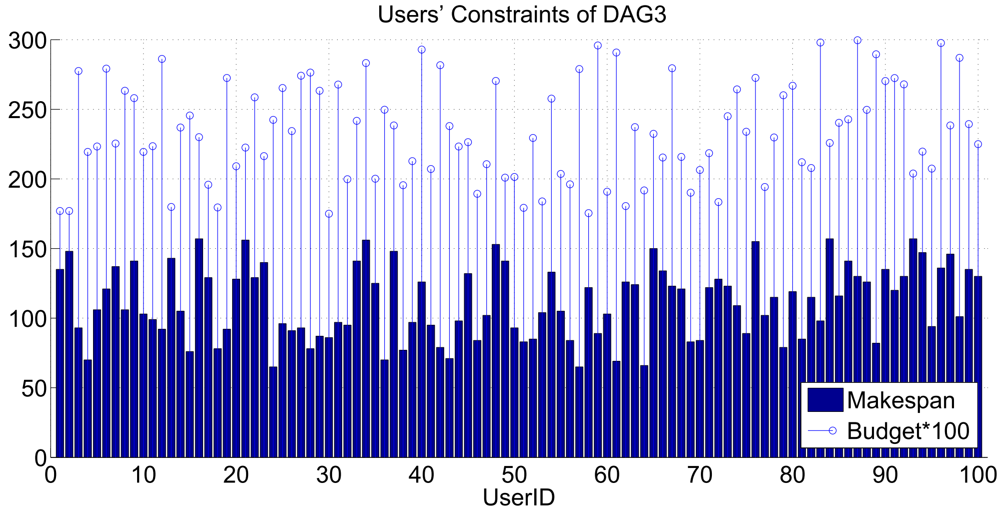


FIGURE 11. Budget&Makespan of tasks of type3.

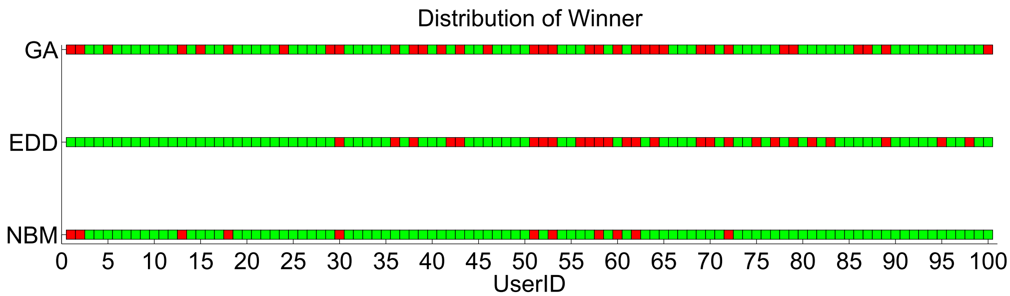


FIGURE 12. Result of tasks of *type*3 using *GA&EDD&NBM*.

TABLE 2. Details of 7 tasks of type3.

DemanderName	Delivery time rank	Budget rank	Bidding price rank
$task^1$	78	97	98
$task^2$	91	98	100
$task^3$	29	15	15
$task^4$	5	63	14
$task^5$	49	59	43
$task^6$	57	13	38
$task^7$	82	56	81

the delivery time is, the smaller the rank value is. The third column is the rank value by budget. The greater the budget is, the smaller the rank value is. The fourth column is the rank value by bidding price. The greater the bidding price is, the smaller the rank value is. For example $task^6$, with a longer delivery time, is at the latter position, the corresponding rank value is 57. But $task^6$ has a greater budget and the corresponding rank value is 13. In EDD algorithm, $task^6$ could not select services until the preceding 56 tasks have been finished. In market economy, $task^6$ don't want to accept this result because it has strong financial support, and it is possible to select services with higher priority by raising the bidding price to prioritize premium cloud services. With full consideration of the demander initiative, task constraints, and economic benefit, the proposed bidding strategy can help user to get a higher priority to service selection. In this experiment, the rank value of $task^6$ is promoted to 38 successfully. Comparing NBM with GA, the $task^5$ cannot be accomplished by using GA to optimize scheduling solution, as shown in Fig. 12. Because it may be the sacrifice for global optimum. However, the demander of $task^5$ is capable of violating the passive scheduling solution by depending on sufficient budget. The proposed algorithm can guarantee participants taking the initiative in resource scheduling.

B. APPLICATION TEST

Crane is a kind of common industrial equipments and is widely used in various fields. Chinese crane industry starts relatively later than abroad, but develops rapidly with the driving of metallurgy, energy, transportation and other basic industries. In recent years, the number of crane manufacturers has soared, and the total value of industry reaches record high. However, the scale and ability of crane manufacturing enterprises are different, manufacturing resources and capabilities are uneven distributed, and can not be shared effectively, resulting in resources insufficiency or serious waste. The ability of industry chain cooperation is low and

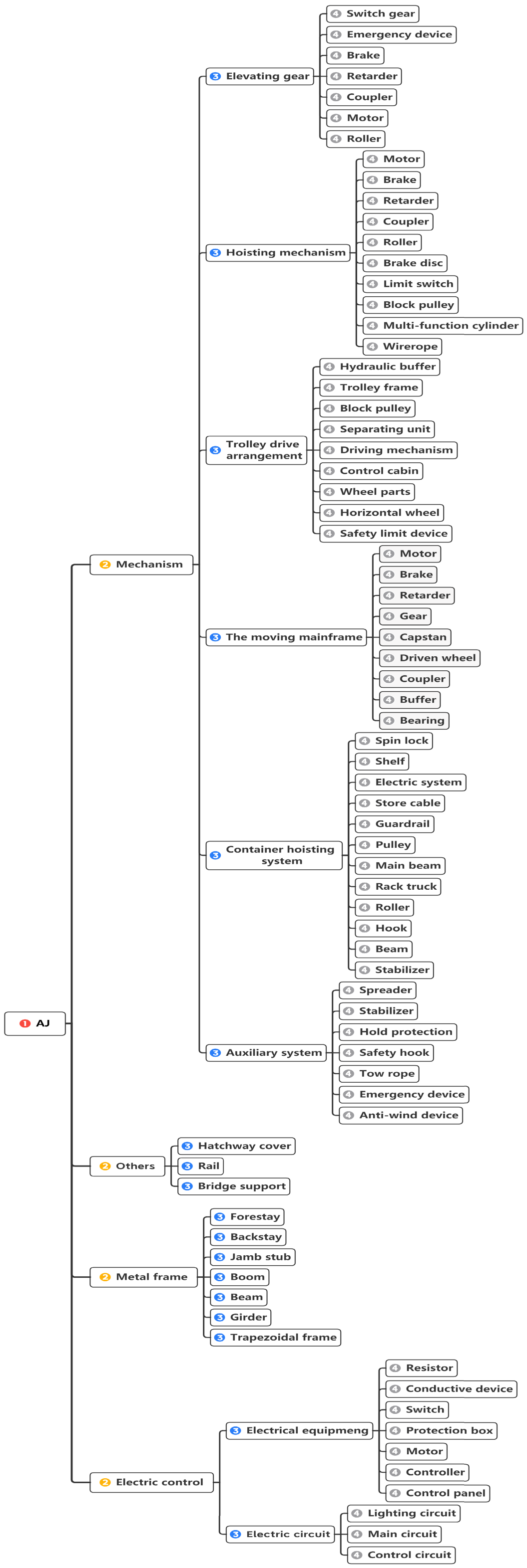


FIGURE 13. Shore crane structure diagram.

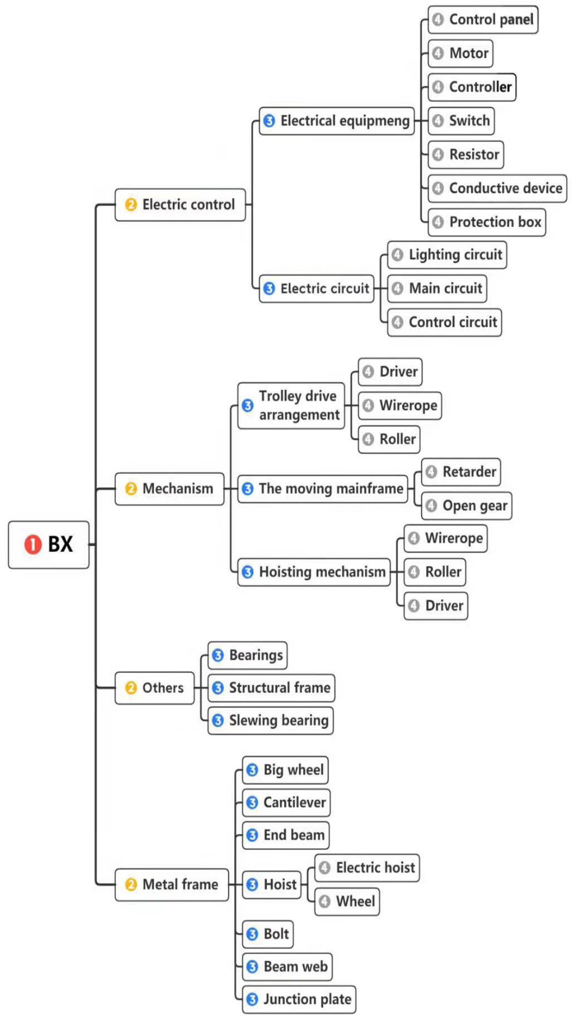


FIGURE 14. Wall sleeving crane structure diagram.

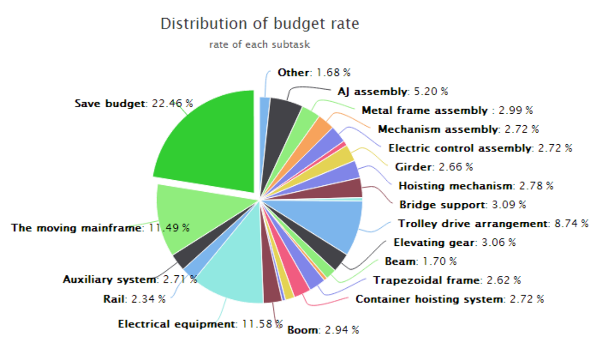


FIGURE 15. Pie of budget distribution.

the product integration is also lower, so the crane enterprises can not quickly respond to the market, the product lacks of market competitiveness. Based on real crane industry, a test version of cloud manufacturing system is developed for crane industry. A test have been done in the system by adopting simulation data and abstract models.

In this test, two types of cranes are taken as examples, shore crane (short: AJ) and wall sleeving crane (short: BX). The cranes can be decomposed into several subtasks as their structures, as shown in Fig. 13 and Fig. 14. Each subtask has

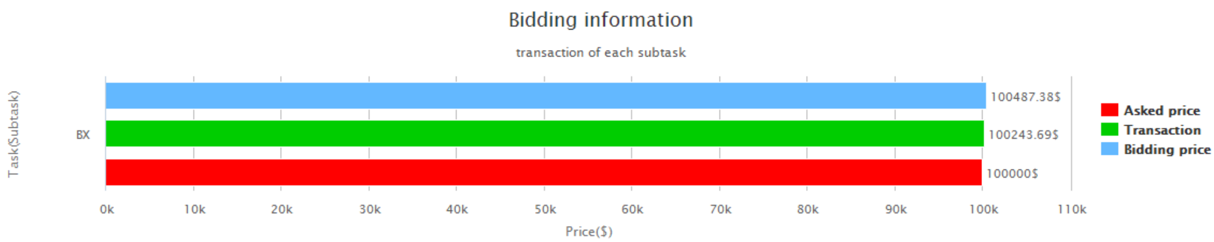


FIGURE 16. Bar of D's trade.

TABLE 3. Details of tasks.

Demander	Task	Budget($)	Delivery	Decomposition level
A	AJ	2000000	2016.5.31	4
B	BX	100000	2016.2.29	2
C	AJ	3000000	2016.4.1	3
D	BX	150000	2016.4.7	1

TABLE 4. Bidding result of tasks.

Demander	Task	Budget($)	Delivery	Bidding result
A	AJ	2000000	2016.5.31	Failure
B	BX	100000	2016.2.29	Failure
C	AJ	3000000	2016.4.1	Success
D	BX	150000	2016.4.7	Success

its own label number, which represents its level of structure. If one task is decomposed, the main contents of subtasks, which need to be further decomposed, would be assembly. At most 4 levels of task decomposition are supported in this test system. Reference historical order data, the cost of AJ and BX are in the interval [2000000$, 3000000$] and [90000$, 150000$], the makespan of AJ are in the interval [80, 200] and [50, 100], the unit is day. Four crane demand orders are simulated as shown in Table 3. Four enterprises, marked A, B, C and D, are in need of cranes. The start time of bidding is 2016.01.01. Decomposition level shows whether the tasks is decomposed and which decomposed level is, 1 means the task does not need to be decomposed. There are almost 2000 service providers and 6000 services supporting the tasks or subtasks of the two types of cranes.

The algorithm test results are shown in Table 4. A and B lose, and C and D win in this bidding mechanism. As for the winners C and D, the details of trades are shown in Table 5 and 6. By statistical the test results, the total cost of C is 2326136.14$, the difference between C's budget and total cost is 673863.86$. It can be calculated from the test result that the proposed algorithm can save the budget by 22.46%, therefore the economic benefit is guaranteed. The scale of the budget for each subtask of C is shown in Fig. 15.

Fig. 16 displays a bar chart of the transaction information of D. The blue represents the bidding price of D, the red represents the asked price of the service, and the green represents the final transaction value. The observation

TABLE 5. Details of C's trades.

Subtask	Bidding price($)	Transaction value($)	Asked price($)
AJ assembly	156144.57	156144.57	155500.00
Metal frame assembly	90264.47	89732.24	89200.00
Mechanism assembly	81611.60	81505.80	81400.00
Electric control assembly	81931.42	81665.71	81400.00
Girder	79828.84	79714.42	79600.00
Hoisting mechanism	83554.06	83277.03	83000.00
Bridge support	92686.93	92643.47	92600.00
Trolley drive arrangement	262800.81	262150.41	261500.00
Elevating gear	92023.12	91911.56	91800.00
Container hoisting system	81474.32	81474.32	81400.00
Beam	51213.17	51056.59	50900.00
Trapezoidal frame	79668.87	78634.44	77600.00
Boom	88211.75	88205.88	88200.00
Electrical equipment	348258.06	347329.03	346400.00
Rail	70287.21	70143.61	70000.00
Auxiliary system	82259.24	81429.62	80600.00
The moving mainframe	345643.39	344821.70	344000.00
Electric circuit	23440.80	23320.40	23200.00
Forestay	15780.09	15640.05	15500.00
Backstay	15913.99	15907.00	15900.00
Hatchway cover	43482.67	43441.34	43400.00
Jamb stud	15685.04	15667.52	15650.00
Others	50638.85	50319.43	50000.00

TABLE 6. Details of D's trades.

Subtask	Bidding price($)	Transaction value($)	Asked price($)
BX	100487.38	100243.69	100000.00

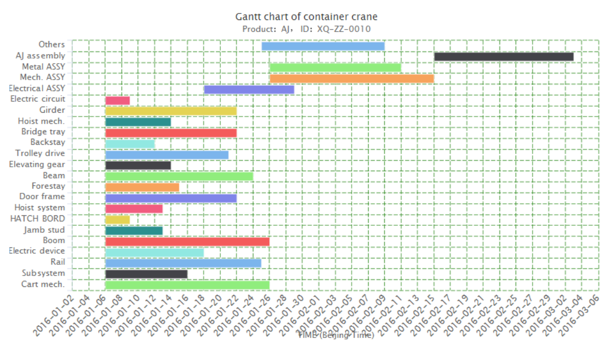


FIGURE 17. Gantt of C's task.

shows that although the budget of D is more than 150000$, there is enough time for it to make a lower bidding price, so D can win by bidding 100487$. The asked price is only 100000$, but the final transaction value is 100243$. The service provider receives higher return than expected. It is proved that the proposed mechanism can effectively guarantee the economic benefit of both demanders and providers.

Fig. 17 shows the execution time for each subtask of the task submitted by C. It is clear that the task is finished in more than 30 days, ahead of the expected deadline, fully meeting the demander requirements.

VI. CONCLUSION

In the market-oriented cloud manufacturing environment, economic benefit is the focus of all the participants. Intention to maximize their own interests will inevitably cause fierce and brutal competition for resources, especially for high-end resources. In this paper, a new fair resource bidding mechanism is proposed in cloud manufacturing. In this algorithm, demanders choose services initiatively by adjusting bidding price depending on their delivery time and financial budget, rather than passively accept the service scheduling scheme. Users don't have to know the complete market information. Only by using private information users can ensure their own economic benefits. The bidding mechanism designed in this paper also breaks through the limitation of the traditional methods that only works in the case of homogeneous resource. Even though the cloud services and manufacturing tasks are heterogeneous, the proposed bidding mechanism is still valid. For decomposable tasks, the demander considers both the sub-constraints and the global constraints when making the bidding strategy for each subtask in the task workflow scheduling. Simulation results show that the proposed algorithm can not only improve the resource utilization, increase the success rate of resource matching, but also protect the economic interests of the participants in the case of incomplete information.

REFERENCES

[1] R. Dobrescu, D. Merezeanu, and S. Mocanu, "Process simulation platform for virtual manufacturing systems evaluation," *Comput. Ind.*, vol. 104, pp. 131–140, Jan. 2019, doi: 10.1016/j.compind.2018.09.008.

[2] A. Gunasekaran, Y. Y. Yusuf, E. O. Adeleye, T. Papadopoulos, D. Kovvuri, and D. G. Geyi, "Agile manufacturing: An evolutionary review of practices," *Int. J. Prod. Res.*, vol. 57, nos. 15–16, pp. 5154–5174, Oct. 2019, doi: 10.1080/00207543.2018.1530478.

[3] B. S. Wang and H. Y. Hao, "Balanced optimal allocation of resources based on hybrid algorithm of ant colony and fish swarm in manufacturing grid," *Int. J. Perform. Eng.*, vol. 13, no. 7, pp. 1123–1131, Nov. 2017, doi: 10.23940/ijpe.17.07.p15.11231131.

[4] Y. Lu and X. Xu, "Cloud-based manufacturing equipment and big data analytics to enable on-demand manufacturing services," *Robot. Comput.-Integr. Manuf.*, vol. 57, pp. 92–102, Jun. 2019, doi: 10.1016/j.rcim.2018.11.006.

[5] Q. D. Wu and Z. J. Wang, "Mass-data-driven intelligent cloud manufacturing for crane," *Syst. Simul. Tech.*, vol. 13, no. 2, pp. 79–82, May 2017, doi: 10.16812/j.cnki.cn31-1945.2017.02.001.

[6] X. Xu, "From cloud computing to cloud manufacturing," *Robot. Comput.-Integr. Manuf.*, vol. 28, no. 1, pp. 75–86, Feb. 2012, doi: 10.1016/j.rcim.2011.07.002.

[7] F. Tao, Y. Cheng, L. D. Xu, L. Zhang, and B. H. Li, "CCIoT-CMfg: Cloud computing and Internet of Things-based cloud manufacturing service system," *IEEE Trans. Ind. Inf.*, vol. 10, no. 2, pp. 1435–1442, May 2014, doi: 10.1109/tii.2014.2306383.

[8] L. Ren, L. Zhang, L. Wang, F. Tao, and X. Chai, "Cloud manufacturing: Key characteristics and applications," *Int. J. Comput. Integr. Manuf.*, vol. 30, no. 6, pp. 501–515, Jun. 2017, doi: 10.1080/0951192x.2014.902105.

[9] L. Zhou, L. Zhang, L. Ren, and J. Wang, "Real-time scheduling of cloud manufacturing services based on dynamic data-driven simulation," *IEEE Trans. Ind. Inf.*, vol. 15, no. 9, pp. 5042–5051, Sep. 2019, doi: 10.1109/tii.2019.2894111.

[10] H. Li, K. Chan, M. Liang, and X. Luo, "Composition of resource-service chain for cloud manufacturing," *IEEE Trans. Ind. Inf.*, vol. 12, no. 1, pp. 211–219, Feb. 2016, doi: 10.1109/tii.2015.2503126.

[11] T. Li, T. He, Z. Wang, and Y. Zhang, "An approach to Iot service optimal composition for mass customization on cloud manufacturing," *IEEE Access*, vol. 6, pp. 50572–50586, 2018, doi: 10.1109/access.2018.2869275.

[12] G. Adamson, L. Wang, M. Holm, and P. Moore, "Cloud manufacturing—A critical review of recent development and future trends," *Int. J. Comput. Integr. Manuf.*, vol. 30, pp. 1–34, Apr. 2015, doi: 10.1080/0951192x.2015.1031704.

[13] P. Lou, L. Yuan, J. Hu, J. Yan, and J. Fu, "A comprehensive assessment approach to evaluate the trustworthiness of manufacturing services in cloud manufacturing environment," *IEEE Access*, vol. 6, pp. 30819–30828, 2018, doi: 10.1109/access.2018.2837664.

[14] C. Yang, S. Lan, W. Shen, G. Q. Huang, X. Wang, and T. Lin, "Towards product customization and personalization in IoT-enabled cloud manufacturing," *Cluster Comput*, vol. 20, no. 2, pp. 1717–1730, Jun. 2017, doi: 10.1007/s10586-017-0767-x.

[15] L. Zhou, L. Zhang, B. R. Sarker, Y. Laili, and L. Ren, "An event-triggered dynamic scheduling method for randomly arriving tasks in cloud manufacturing," *Int. J. Comput. Integr. Manuf.*, vol. 31, no. 3, pp. 318–333, Mar. 2018, doi: 10.1080/0951192x.2017.1413252.

[16] C. Xie, H. Cai, L. Xu, L. Jiang, and F. Bu, "Linked semantic model for information resource service toward cloud manufacturing," *IEEE Trans. Ind. Inf.*, vol. 13, no. 6, pp. 3338–3349, Dec. 2017, doi: 10.1109/tii.2017.2723501.

[17] M. Y. Kovalyov and E. Pesch, "A game mechanism for single machine sequencing with zero risk," *Omega*, vol. 44, pp. 104–110, Apr. 2014, doi: 10.1016/j.omega.2013.11.001.

[18] P. Zhang, X. Wang, and M. Zhou, "Multi-user multi-provider resource allocation in cloud computing," in *Proc. IEEE 14th Int. Conf. Autom. Sci. Eng. (CASE)*, Munich, Germany, Aug. 2018, pp. 1428–1433, doi: 10.1109/coase.2018.8560365.

[19] G. Xiong, F.-Y. Wang, T. R. Nyberg, X. Shang, M. Zhou, Z. Shen, S. Li, and C. Guo, "From mind to products: Towards social manufacturing and service," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 1, pp. 47–57, Jan. 2018, doi: 10.1109/jas.2017.7510742.

[20] M. Yuan, K. Deng, W. Chaovalitwongse, and S. Cheng, "Multi-objective optimal scheduling of reconfigurable assembly line for cloud manufacturing," *Optim. Methods Softw.*, vol. 32, no. 3, pp. 581–593, May 2017, doi: 10.1080/10556788.2016.1230210.

[21] Q. Fan and N. Ansari, "On cost aware cloudlet placement for mobile edge computing," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 4, pp. 926–937, Jul. 2019, doi: 10.1109/jas.2019.1911564.

[22] X. Li, J. Wan, H.-N. Dai, M. Imran, M. Xia, and A. Celesti, "A hybrid computing solution and resource scheduling strategy for edge computing in smart manufacturing," *IEEE Trans. Ind. Inf.*, vol. 15, no. 7, pp. 4225–4234, Jul. 2019, doi: 10.1109/tii.2019.2899679.

[23] H. Yuan, J. Bi, M. Zhou, and A. C. Ammari, "Time-aware multi-application task scheduling with guaranteed delay constraints in green data center," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 3, pp. 1138–1151, Jul. 2018, doi: 10.1109/tase.2017.2741965.

[24] K. Liu, H. Jin, J. Chen, X. Liu, D. Yuan, and Y. Yang, "A compromised-time-cost scheduling algorithm in SwinDeW-C for instance-intensive cost-constrained workflows on a cloud computing platform," *Int. J. High Perform. Comput. Appl.*, vol. 24, no. 4, pp. 445–456, Nov. 2010, doi: 10.1177/1094342010369114.

[25] J. Zhou and X. Yao, "A hybrid artificial bee colony algorithm for optimal selection of QoS-based cloud manufacturing service composition," *Int. J. Adv. Manuf. Technol.*, vol. 88, nos. 9–12, pp. 3371–3387, Feb. 2017, doi: 10.1007/s00170-016-9034-1.

[26] W. Li, C. Zhu, L. T. Yang, L. Shu, E. C.-H. Ngai, and Y. Ma, "Subtask scheduling for distributed robots in cloud manufacturing," *IEEE Syst. J.*, vol. 11, no. 2, pp. 941–950, Jun. 2017, doi: 10.1109/jsyst.2015.2438054.

[27] L. Zhou, L. Zhang, Y. Laili, C. Zhao, and Y. Xiao, "Multi-task scheduling of distributed 3D printing services in cloud manufacturing," *Int. J. Adv. Manuf. Technol.*, vol. 96, nos. 9–12, pp. 3003–3017, Jun. 2018, doi: 10.1007/s00170-017-1543-z.

[28] L. Zhou, L. Zhang, C. Zhao, Y. Laili, and L. Xu, "Diverse task scheduling for individualized requirements in cloud manufacturing," *Enterprise Inf. Syst.*, vol. 12, no. 3, pp. 300–318, Mar. 2018, doi: 10.1080/17517575.2017.1364428.

[29] M. H. Ghahramani, M. Zhou, and C. T. Hon, "Toward cloud computing QoS architecture: Analysis of cloud systems and cloud services," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 1, pp. 6–18, Jan. 2017, doi: 10.1109/jas.2017.7510313.

[30] J. Bi, H. T. Yuan, W. Tan, M. C. Zhou, Y. S. Fan, J. Zhang, and J. Q. Li, "Application-aware dynamic fine-grained resource provisioning in a virtualized cloud data center," *IEEE T. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 1172–1183, Apr. 2017, doi: 10.1109/TASE.2015.2503325.

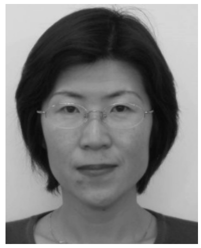
[31] W. Peng, W. Guo, L. Wang, and R.-Y. Liang, "Dynamic pricing in cloud manufacturing systems under combined effects of consumer structure, negotiation, and demand," *Math. Problems Eng.*, vol. 2017, pp. 1–15, 2017, doi: 10.1155/2017/2073585.

[32] H. M. Fard, R. Prodan, and T. Fahringer, "A truthful dynamic workflow scheduling mechanism for commercial multicloud environments," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1203–1212, Jun. 2013, doi: 10.1109/tpds.2012.257.

[33] A. Nezarat and G. Dastghaibifard, "Efficient Nash equilibrium resource allocation based on game theory mechanism in cloud computing by using auction," *PLoS ONE*, vol. 10, no. 10, Oct. 2015, Art. no. e0138424, doi: 10.1371/journal.pone.0138424.

[34] M. J. W. Cock, "The assessment of preference," *J. Anim. Ecol.*, vol. 47, no. 3, pp. 805–816, Oct. 1978, doi: 10.2307/3672.

ZHAOHUI LIU received the M.Eng. degree in guidance, navigation, and control from Northeastern University, China, in 2014. He is currently pursuing the Ph.D. degree in control science and engineering with Tongji University, China. His research interests primarily include game theory, evolutionary algorithms, networked manufacturing, optimization resource scheduling, mechanism design, multiagent systems, complex systems, and pinning control.



ZHONGJIE WANG (Member, IEEE) received the Ph.D. degree in control theory and control engineering from Tongji University, Shanghai, China, in 2000. From 2000 to 2002, she was a Postdoctoral Program Fellow in control science and engineering, Tongji University, where she joined the Department of Control Science and Engineering, in 2002. She was an Associate Professor, from 2002 to 2006. She has been a Professor, since 2006. She has authored or coauthored more than 80 articles in refereed international journals and conferences. Her research interests involve smart manufacturing, data analysis, intelligent decision, and AI application.