Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 369350, 9 pages
http://dx.doi.org/10.1155/2014/369350

Research Article

Multitask Oriented Virtual Resource Integration and Optimal Scheduling in Cloud Manufacturing

Zhen Cheng,[1] Dechen Zhan,[1] Xibin Zhao,[2] and Hai Wan[2]

[1] *School of Computer Science and Technology, Harbin Institute of Technology, 92 West Dazhi Street, Nangang, Harbin 150001, China*
[2] *School of Software, Tsinghua University, NLIST, KLISS, Beijing 100084, China*

Correspondence should be addressed to Zhen Cheng; chengzhen_hit@163.com

Received 13 February 2014; Accepted 2 April 2014; Published 4 May 2014

Academic Editor: Xiaoyu Song

To deal with the problem of resource integration and optimal scheduling in cloud manufacturing, based on the analyzation of the existing literatures, multitask oriented virtual resource integration and optimal scheduling problem is presented from the perspective of global optimization based on the consideration of sharing and correlation among virtual resources. The correlation models of virtual resources in a task and among tasks are established. According to the correlation model and characteristics of resource sharing, the formulation in which resource time-sharing scheduling strategy is employed is put forward, and then the formulation is simplified to solve the problem easily. The genetic algorithm based on the real number matrix encoding is proposed. And crossover and mutation operation rules are designed for the real number matrix. Meanwhile, the evaluation function with the punishment mechanism and the selection strategy with pressure factor are adopted so as to approach the optimal solution more quickly. The experimental results show that the proposed model and method are feasible and effective both in situation of enough resources and limited resources in case of a large number of tasks.

1. Introduction

Cloud manufacturing is a typical model of service-oriented manufacturing [1]; its core idea is to form the pool of cloud manufacturing resource based on the virtualization for manufacturing resources and servitization for manufacturing capacity through the integration of new technologies such as cloud computing and IoT and then provide various manufacturing services which can be obtained anytime, used on demand, safe and reliable, of high-quality, and of low-cost for the full life-cycle manufacturing process, and finally to achieve customers, manufacturers, and providers value added jointly [2, 3].

In cloud manufacturing service system, based on cloud manufacturing service platform, manufacturing enterprises can integrate a large number of personalized requirements to form clustering requirements and can form a dynamic entity named cloud enterprise [4] according to dynamic configuration of clustering requirements and integration of virtual manufacturing resources from the pool of cloud manufacturing resources. The cloud enterprise can provide products or services on demand and finally can satisfy personalized requirements through the optimal scheduling of virtual resources and the mapping of virtual resources and real resources.

After receiving customers' requirements, cloud manufacturing enterprise usually fulfills through several steps: (1) task classification; (2) task decomposition; (3) resource allocation; (4) resource optimal scheduling; (5) task execution and services feedback evaluation. A large number of existing literatures are based on single task [5–8]; however, tasks are received continuously in reality. If the method for single task problem is used to solve multitask problem, there would be some following shortages that the scheduling result makes every single task optimized but might not be the best in global view.

(1) If multiple tasks are scheduled one by one adopting the existing single task oriented scheduling method simply, we should sort tasks using typical strategy (e.g., first-come-first-service, task priority marked, etc.) firstly and then configure

optimal schedule resources for each task in turn. This makes the optimization goal that the configuration and optimal scheduling of resources only meet the current task each time. The problem is that if the key resources can meet lots of tasks, but only used in the priority task, this will lead to the optimization for multiple tasks from a global perspective that cannot be achieved. Even though the literature [9] optimized the problem from the global view, it did not consider the relationship of resources which would be emphasized in this paper.

(2) Only the exclusivity of resources is considered, and the sharing of resources is not taken into account in a scheduling process. The exclusivity of resources means that if a resource is occupied or used, it cannot be used by other tasks at the same time. In the related literatures of service computing and/or cloud computing [9, 10], the exclusivity of resources means that a resource is used only once in a scheduling process, no matter how many tasks are in a scheduling process. And also the exclusivity of resources is a typical characteristic of resources scheduling in service computing and/or cloud computing environment, because the unit of time for a scheduling is usually seconds or even milliseconds, microseconds [11, 12]. However, in cloud manufacturing environment, the unit of time for resources scheduling is usually days or weeks, so in order to enhance the resources utilization, resources should be considered to reuse in other tasks when resources are free in a multiple tasks scheduling process. In this paper, we call the time division sharing of resources.

(3) The optimal scheduling did not consider the correlation of resources. Resource scheduling, service selection, discovery, and composition in the literatures are based on the features of resources and/or services [13–15]. However, in fact, there are correlations among resources, particularly resources were used in tasks. Normally, for example, two consecutive subtasks in a task performed by resources of the same provider should be more efficient than different providers. Here, the same provider is the correlation of resources.

In cloud manufacturing environment, tasks may be continuous, but, in this paper, we put time into segments and handle all tasks in a time slot just like planning in manufacturing enterprises. And then we could optimally schedule virtual resources at the beginning of a time slot, fully considering the correlation and sharing of resources, to break through the limitation of local optimization and to achieve the overall optimization for multiple tasks.

2. Problem Formulation

Just like service composition and optimal scheduling for single task, every single task of multiple tasks for optimal scheduling in cloud manufacturing also should go through decomposition, production of candidate resource sets, selection of resources, and selection of execution path. From the perspective of the execution path of subtasks and candidate resource sets, tasks will be divided into 3 categories in this paper.

(1) Different types of tasks: the execution path of subtasks and corresponding candidate resource sets are totally different. For example, tasks from logistics enterprises and tasks from manufacturing enterprises are different types.

(2) Same type of tasks: the execution path of subtasks and corresponding candidate resource sets are totally the same. For example, multiple tasks from a certain type of product manufacturing enterprises are the same type.

(3) Mixed types of tasks: the execution path of subtasks and corresponding candidate resource sets are partly the same. For example, multiple tasks from different types of product manufacturing enterprises are mixed types. All the tasks have different manufacturing processes but require the same cutting process.

Virtual resources integration and optimal scheduling for different types of tasks could adopt the method of traditional single task execution because of totally different candidate resource sets and execution paths. And for mixed types of tasks, we will study it in the future because of its complexity. So in this paper, we will focus on the same type of tasks and study virtual resource integration and optimal scheduling.

2.1. Motivating Example. Before introducing the problem, an example of manufacturing process of C70E open wagon in CNR is presented to illustrate the problems that are addressed in this paper. The task of manufacturing process for C70E open wagon, denoted by $Ta(\mathit{deliverTime, price})$, has five subtasks such as wheel processing, hook lock processing, body processing, assembling, painting, experiment, and delivery. Assuming the candidate set of virtual resources (VRS) for each process has three candidate virtual resources (VR), as shown in Figure 1, the first number of the candidate resource denotes the processing days and the second number denotes the cost. For example, the first number “5” of virtual resource $VR_1^1(5, 700)$ indicates the processing days and the second number “700” indicates the cost.

When a cloud manufacturing enterprise received 3 tasks: $\{Ta_1(30, 5000), Ta_2(25, 8000), Ta_3(15, 10000)\}$, we can get an optimal execution path of resource integration $\{P_1, P_2, P_3\}$ to maximize profit Max $\sum_{i=1}^{3}$ Profit(P_i), using traditional heuristic method without considering the sharing and correlation of resources. However, when the cloud manufacturing enterprise receives the fourth task $Ta_4(20, 9000)$ at the same time, the tasks could not be fully scheduled without considering the sharing of resources. The general method is that we could select 3 tasks $\{Ta_2, Ta_3, Ta_4\}$ from the 4 tasks to maximize profit. But if we consider the sharing of resources, the cloud manufacturing enterprise could fully schedule the 4 tasks

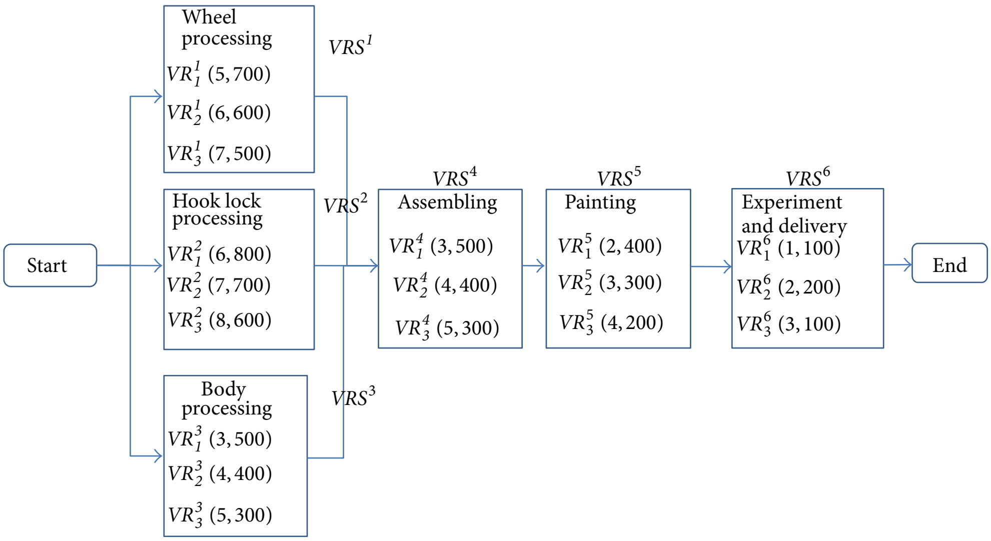

Figure 1: Manufacturing process of C70E open wagon.

before the deadline of each task. The following is a feasible solution (maybe not the best solution):

$$
\begin{gathered}
P_1 \left\langle \left\langle VR_3^1, VR_3^2, VR_3^3 \right\rangle, VR_3^4, VR_3^5, VR_3^6 \right\rangle \\
P_2 \left\langle \left\langle VR_2^1, VR_2^2, VR_2^3 \right\rangle, VR_2^4, VR_2^5, VR_2^6 \right\rangle \\
P_3 \left\langle \left\langle VR_1^1, VR_1^2, VR_1^3 \right\rangle, VR_1^4, VR_1^5, VR_1^6 \right\rangle \\
P_4 \left\langle t_{\text{wait}}^4, \left\langle VR_1^1, VR_1^2, VR_1^3 \right\rangle, VR_1^4, VR_1^5, VR_1^6 \right\rangle,
\end{gathered}
\tag{1}
$$

where $t_{\text{wait}}^4 = \text{MAX}(VR_1^1, VR_1^2, VR_1^3) = 6$. The performance period of tasks is shown in Figure 2.

The feasible solution is that we just delayed starting the fourth task Ta_4 6 days ($t_{\text{wait}}^4 = 6$). The feasible solution is typically considering the time division sharing of resources ($VR_1^1, VR_1^2, VR_1^3, VR_1^4, VR_1^5, VR_1^6$). Obviously, the feasible solution could be further optimized.

On the other hand, there is correlation among resources from the manufacturing and service perspective. In this paper, we divided the correlation of resources into two categories.

(1) The correlation of resources in a task: we take the first task Ta_1, for example, if the selected resources of the last three subtasks belong to the same provider ($VR_3^4 \cdot \text{provider} = VR_3^5 \cdot \text{provider} = VR_3^6 \cdot \text{provider}$), the cost of VR_3^4, VR_3^5, VR_3^6 will be lower than the default cost. If all of them have 10% off, the cost of execution path P_1 will decrease from 2000 to 1940.

(2) The correlation of resources among tasks: we take the feasible solution, for example, the resources $VR_1^1, VR_1^2, VR_1^3, VR_1^4, VR_1^5, VR_1^6$ are used twice, and then the cost will be reduced per use. Assume that all of them have 5% off and the cost of path P_3 will be reduced 150, the same as P_4. Finally, the cost of solution will be reduced by 300 than without correlation among tasks.

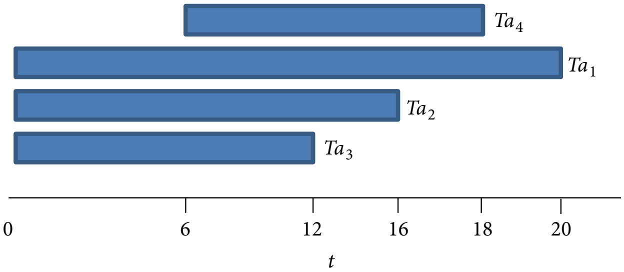

Figure 2: Multitask performance period.

So, the sharing and correlation of resources are fully considered to realize virtual resource integration and optimal scheduling for multitask in this paper.

2.2. Problem Formulation

2.2.1. Resource Correlation Modelling. For the correlation in a task, we use ontology to describe virtual resources firstly. The virtual resource model can be defined as a 3-tuple: $VR(TRAS, CRAS, OAS)$, where $TRAS$ denotes time related attributes of virtual resources, for example, capacity and resource provider. $CRAS$ denotes cost related attributes of virtual resources, for example, resource provider and the location of corresponding resources. OAS denotes the other attributes, for example, the size of corresponding resources. In this paper, we realize virtual resources integration and optimal scheduling for multitask from two aspects: time and cost. For the other attributes such as the reliability of virtual resources, we could also handle using the proposed method.

Secondly, we can model the correlation of resources based on the virtual resource model. We take the cost of virtual resources $C(VR)$ as an example to model the correlation of resources in a task, and the same as the time of virtual resources $T(VR)$:

$$C(VR) = \left\{\text{default} : \text{Value}_0; g_1\left(\overline{VRS}.\text{selected}.CRA\right) : \text{Value}_1; g_2\left(\overline{VRS}.\text{selected}.CRA\right) : \text{Value}_2; \ldots\right\}, \quad (2)$$

where Value_0 denotes the default value without considering the correlation and $\overline{VRS}$ denotes the other candidate virtual resource sets in which the virtual resource VR is not included. CRA denotes the cost related attributes; the Boolean function g is the correlation function of resources. Therefore, $g_1(\overline{VRS}\cdot \text{selected}\cdot CRA) : \text{Value}_1$ denotes that if the CRA of the selected resource in the $\overline{VRS}$ makes the Boolean function g_1 true, the value of the $C(VR)$ will be Value_1. Obviously, the Boolean function g has recursive operations; we only use NOT and AND operations as follows:

$$\begin{gathered} g_i() = \overline{g_j()}, \quad i \neq j; \\ g_i() = g_j() \,\&\&\, g_k(), \quad i \neq j \neq k. \end{gathered} \quad (3)$$

For the OR and XOR operations, we can use NOT and AND operations to realize. And NOT operation is prior to AND operation. If there are multiple correlation functions to meet at the same time, the value of the $C(VR)$ will be selected as the minimized value because the smaller the cost is the better. For other cases, such as reliability, it is the bigger the better.

Therefore, in this paper, the calculation model for time and cost based on resource correlation in a task is as follows:

$$T(VR) = \text{Min}\left\{\text{Value}_0, \text{Value}_i\right\}, \quad \text{where } g_i() = \text{true}. \quad (4)$$

$$C(VR) = \text{Min}\left\{\text{Value}_0, \text{Value}_i\right\}, \quad \text{where } g_i() = \text{true}; \quad (5)$$

We will take a detailed example to illustrate the correlation of resources in a task. We still take the cost of VR_3^4 in the feasible solution P_1 as an example to model the correlation. Firstly, we build the ontology description model of VR_3^4 as follows: $VR_3^4(\{\cdots\}, \{CNR, QiQiHaer, \ldots\}, \{\cdots\})$, where the first set is $TRAS$, and the third set is OAS. The second set is the value of $CRAS$; that is to say the provider of VR_3^4 is CNR, and the location is $QiQiHaer$. Then, we can build the cost calculation model $C(VR_3^4)$ as follows:

$$\begin{aligned} C\left(VR_3^4\right) = \{&\text{default} : 300; \\ &VRS^5.\text{selected.provider} = CNR : 285; \\ &VRS^5.\text{selected.provider} \\ &= CNR \,\&\&\, VRS^6.\text{selected.provider} \\ &= CNR : 270; VRS^5.\text{selected.location} \\ &= QiQiHaer : 295 \ldots\}. \end{aligned} \quad (6)$$

In the above model, the default cost of using VR_3^4 is 300. If the provider of the selected resources of VRS^5 corresponding to the fifth subtask is also CNR, the cost of using VR_3^4 will be 285. And further, if the provider of the sixth subtask is also CNR, the cost of using VR_3^4 will be 270. If the location of the selected resources of VRS^5 corresponding to the fifth subtask is $QiQiHaer$, the cost of using VR_3^4 will be 295. Obviously, if the provider of the fifth subtask is the same as the sixth subtask and also is the CNR, the second and the third correlation functions in the above model are met at the same time. According to the features of cost, the minimized value 270 (the third correlation function) will be selected.

For the correlation among tasks, we also take the cost of virtual resources $C(VR)$ as an example to model the correlation of resources among tasks, and the same as the time of virtual resources $T(VR)$:

$$\begin{aligned} C(VR) = \{&\text{default} : \text{defaultValue}; \\ &\text{correlationFunc}_1(\text{DefaultValue}) : \text{Value}_1; \\ &\text{correlationFunc}_2(\text{DefaultValue}) : \text{Value}_2; \ldots\}, \end{aligned} \quad (7)$$

where DefaultValue is calculated by formula (5). $\text{correlationFunc}_1(\text{DefaultValue}) : \text{Value}_1$ denotes that if the selected resources in a scheduling process satisfy the correlation function correlationFunc_1, the value of the $C(VR)$ will be Value_1. For example, the cost per use may be reduced according to the increased times of use. And then, we can simply define the correlationFunc_1 as follows:

$$\text{correlationFunc}_1 = \text{DefaultValue} * \text{Dec}(VR)^{\text{Num}(VR)-1}, \quad (8)$$

where $\text{Num}(VR)$ is the times of using VR and $\text{Dec}(VR)$ denotes cost decay rate.

Again, in this paper, the calculation model for time and cost based on resource correlation among tasks is as follows:

$$\begin{gathered} T(VR) = \text{Min}\left\{\text{Value}_0, \text{Value}_i\right\}, \\ C(VR) = \text{Min}\left\{\text{Value}_0, \text{Value}_i\right\}. \end{gathered} \quad (9)$$

2.2.2. Problem Formulation. Taking account of all the above features, we give the problem description as follows.

Let $Ta = \{Ta_1, Ta_2, \ldots, Ta_n\}$ denote the same type of tasks which the cloud manufacturing enterprise receives within a period of time. Each task has corresponding bid price $\{\text{Price}_1, \text{Price}_2, \ldots, \text{Price}_n\}$. Let $\{T_1, T_2, \ldots, T_n\}$ denote the number of days to delivery date for each task. And each task can be decomposed to J subtasks. Let $P = \{P_1, P_2, \ldots, P_m\}$ be a feasible solution. Each subtask can be executed by a resource of the corresponding candidate resource set VRS_j^i. And there are J_K resources $\{{}^{1}VR_j^i, {}^{2}VR_j^i, \ldots, {}^{K}VR_j^i\}$ in each candidate resource set VRS_j^i. Then the feasible solution should be under the following constraints:

$$m \leq n \tag{10}$$

$$T\left(P_i\right) \leq T\left(Ta\left(P_i\right)\right) \tag{11}$$

$$\text{if } {}^{K}VR_j^i\left[t_1, t_2\right] \in P_i, \qquad {}^{K}VR_j^i\left[t_1, t_2\right] \notin \bigcup P_p, \quad i \neq p; \quad 1 \leq p \leq m. \tag{12}$$

The constraint (10) means that some tasks may be not executed. In constraint (11), $Ta(P_i)$ denotes the task corresponding to the feasible solution P_i. And constraint (11) means that all the executed tasks must be finished before delivery date. Constraints (12) mean that the exclusivity and sharing are considered for each selected resource. If the ${}^{K}VR_j^i$ was used by P_i in the period of $[t_1, t_2]$, it cannot be used in any other P_p $(i \neq p)$ at the same time, but it can be used in any other period.

How to evaluate the feasible solution? From the perspective of cloud manufacturing enterprise, the following objects should be considered.

(i) The more the profit the better:

$$\text{MAX} \sum_{i=1}^{m} \text{Profit}\left(P_i\right), \tag{13}$$

where $\text{Profit}(P_i) = \text{Price}(Ta(P_i)) - C(P_i)$.

(ii) The bigger the number of tasks that can be executed the better:

$$\text{MAX}\,(m). \tag{14}$$

The above two objects are not necessarily linear relations. For an extreme example, the profit of a task may be more than all the other tasks in an enterprise. In reality, it is also meaningful. For example, a start-up cloud manufacturing enterprise may reduce some profits to get more customers, then get more market share, and finally get more profits.

2.3. Model Analysis. Obviously, the problem of multitask oriented virtual resource integration and optimal scheduling in cloud manufacturing belongs to the multiobjective optimization problem. Such problems can usually be solved by Pareto optimal method and can also be transformed to single objective to solve by weighting method. In this paper, we transform the multiple objectives to single objective as follows by linear weighting method:

$$\text{MAX}\left(\alpha * \sum_{i=1}^{m} \text{Profit}\left(P_i\right) + \beta * m\right), \tag{15}$$

where α and β are the corresponding weighted parameters to control the relative significance of each objective, and $\alpha + \beta = 1$. The values will be given according to the target of the optimization.

In order to reduce the number of variables in the model and obtain more convenience to solve the problem, we normalize and handle the $\text{Profit}(P_i)$ as follows:

$$\overline{\text{Profit}\left(P_i\right)} = \frac{\text{Profit}\left(P_i\right) - \text{Min}\left(\text{Profit}\left(P_i\right)\right)}{\text{Max}\left(\text{Profit}\left(P_i\right)\right) - \text{Min}\left(\text{Profit}\left(P_i\right)\right)}. \tag{16}$$

The variables $X = (x_1, x_2, \ldots, x_n)$ are given, where x_i is Boolean value. If the value of x_i is 1, the task Ta_i will be executed. Otherwise, the task Ta_i will not be executed. And then (15) can be transformed as follows:

$$\text{MAX}\left(\alpha * \sum_{i=1}^{n}\left(\overline{\text{Profit}\left(Ta_i\right)} * x_i\right) + \beta * \sum_{i=1}^{n} x_i\right). \tag{17}$$

Therefore, (11) can be transformed as follows:

$$T\left(P_i\right) * x_i \leq T_i * x_i. \tag{18}$$

In order to simplify (12), it is assumed that we will schedule tasks in the future period T and divide T into several equal parts. Therefore, in any period $t \in T$, we will give the Boolean variable $y(i, j, k, t) \cdot y(i, j, k, t)$ which denotes that if the kth virtual resource in the candidate resource set VRS_j^i is selected to execute the task Ta_i in the period t, the variable $y(i, j, k, t)$ will be true. Otherwise, the variable $y(i, j, k, t)$ will be false. And then (12) will be transformed as follows:

$$\sum_{i=1}^{n} y\left(i, j, k, t\right) \leq 1, \quad \forall t \in T, j, k \text{ fixed}. \tag{19}$$

Meanwhile, the variable $y(i, j, k, t)$ should satisfy the following constraint to ensure that a subtask is only executed by a resource in the period t:

$$\sum_{k=1}^{J_k} y\left(i, j, k, t\right) \leq 1, \quad \forall t \in T, i, j \text{ fixed}. \tag{20}$$

Therefore, the mathematical formulations for multitask oriented virtual resource integration and optimal scheduling in cloud manufacturing are as follows: (17) subject to (18), (19), (20), and (21). Consider that

$$x_i \in \{0, 1\}, \quad 1 \leq i \leq n. \tag{21}$$

Apparently, the above problem belongs to NP-hard problem.

3. Virtual Resource Integration and Optimal Scheduling

GA is a population-based stochastic optimization technique inspired by the mechanism of natural selection and evolution. According to the code of solution space and the operation of the decision variables for the problem, GA can get the proximate optimal solution quickly by guided search in the total solution space. Therefore, GA is usually used to solve the problem of composition and optimal scheduling under the complex constraints.

In this paper, we adopt GA based on the real number matrix code to solve the problem of multitask oriented virtual resource integration and optimal scheduling in cloud manufacturing.

```
(1) For 1 to $T \times J$ // looped by columns
(2)   For 1 to $n$ //checked by rows
(3)     If $a_{ijt} = 0$, jump to Step 2;
(4)     Else
          If $VC_i = 0$, jump to Step 2;
          Else
            If not existed the same non-zero number, jump to
            Step 2;
            Else
              Randomly selected a number from the difference
              set between the $[0, J_k]$ set and the current column
              value set to replay. If the difference sets is Ø, then
              set $a_{ijt} = 0$;
(5)   For End
(6) For End
```

Algorithm 1

3.1. Chromosomal Representation. The solution is coded as the following real number matrix:

$$A = \begin{bmatrix} a_{111} & a_{121} & \cdots & a_{1J1} \cdots a_{11T} & a_{12T} & \cdots & a_{1JT} \\ a_{211} & a_{221} & \cdots & a_{2J1} \cdots a_{21T} & a_{22T} & \cdots & a_{2JT} \\ \cdots & \cdots & \cdots & \cdots\cdots\cdots & \cdots & \cdots & \cdots \\ a_{n11} & a_{n21} & \cdots & a_{nJ1} \cdots a_{n1T} & a_{n2T} & \cdots & a_{nJT} \end{bmatrix}. \quad (22)$$

Each row of matrix A denotes a task. The columns in matrix A are divided into T parts, each part indicates a period t, and, in each part, there are J columns which denote J subtasks in each task. Therefore, there are $T \times J$ columns in the matrix A, and the subscript meanings of the element a_{ijt} in the matrix A are as follows:

i: the identification of the ith task;

j: the jth subtask of the task;

t: the tth period.

The value of the matrix element a_{ijt} is an integer from $[0, J_k]$. It means that the a_{ijt}th virtual resource in the jth candidate resource set is selected to execute the task Ta_i in the period t where $a_{ijt} \neq 0$. Otherwise, the jth subtask of the task Ta_i in the period t will not be executed.

The matrix is related to the variables as follows.

(i) If the values of the ith row in the matrix are all 0, the ith task Ta_i will not be executed, and then the variable x_i equals 0. Otherwise, the value of the variable x_i is 1.

(ii) If $a_{ijt} \neq 0$ in the period t, the variable $y(i, j, a_{ijt}, t)$ equals 1, and then the values of all the other rows in the jth column of the tth part cannot equal a_{ijt} which is expressed as $y(i', j, a_{ijt}, t) = 0$, where $i' \neq i$. Obviously, the constraint (19) is satisfied.

(iii) If $a_{ijt} \neq 0$ in the period t, the variable $y(i, j, a_{ijt}, t)$ equals 1, and then for the same i and j, $y(i, j, k, t) = 0$, where $k \neq a_{ijt}$ and $k \in [0, J_k]$. Obviously, the constraint (20) is satisfied.

3.2. Initial Population. Each individual in the initial population is generated randomly column by column as follows so as to ensure the individual randomness and avoid the local optimization.

(i) In the period of $t = 0$, we used the following method to generate the first column value. If the number of tasks is less than the number of virtual resources in the corresponding candidate resources set, that is, $n \leq J_k$, we will randomly select n values from $[0, J_k]$ to form a random sequence as the first column. Otherwise, we will select all of the J_k values and $n - J_k$ zeros to form a random sequence as the first column. After that the value of all the other $J - 1$ columns in the period of $t = 0$ will be zero according to the rule that any task can be transformed to the sequence process [13].

(ii) In the period of $t > 0$, we firstly see the corresponding column of the period $t - 1$. If the jth subtask is not finished in the period $t - 1$, then $a_{ijt} = a_{ij(t-1)}$. If the subtask is finished, then $a_{ijt} = 0$ and $a_{i(j+1)t} \neq 0$ at the same time. The value of the $a_{i(j+1)t}$ should be randomly picked from the remaining numbers after being finished by $a_{i(j+1)t} = a_{i(j+1)(t-1)}$.

3.3. Crossover and Mutation. Because of the real number encoding of the solution, the traditional crossover and mutation operators of the Boolean value cannot be simply used in this paper. We design the crossover and mutation operators as follows.

(1) Row-Based Crossover. We will firstly generate an n-dimension Boolean type column vector VC. We will swap the ith row of the two parent matrices if $VC_i = 1$, do nothing if $VC_i = 0$, and finally generate the new two matrices. The constraints may not be satisfied, that there may be the same nonzero number in the same column after row-based crossover operating. Therefore, we must run the following fixed program to make the new two matrices satisfy the constraints. See Algorithm 1.

(2) Column-Based Crossover. Similar to the row-based crossover, we will firstly generate a $T \times J$-dimension Boolean

type row vector VC. We will swap the jtth row of the two parent matrixes if $VC_{jt} = 1$, do nothing if $VC_{jt} = 0$, and finally generate the new two matrixes. Obviously, the new two matrixes also satisfy the constraints.

(3) Block-Based Crossover. Block-based crossover is operated for the unit of block corresponding to the period of t. Therefore, we will generate a T-dimension Boolean type row vector VC. We will swap the tth block of the two parent matrixes if $VC_t = 1$, do nothing if $VC_t = 0$, and finally generate the new two matrices. Obviously, similar to the column-based crossover, the new two matrices also satisfy the constraints.

(4) Row-Based Mutation. Firstly, we give the mutation rules. We will change the value as follows if the value is not zero and do not change the value if the value is zero:

$$a_{ijt} = J_k - a_{ijt} + 1. \tag{23}$$

For the row-based mutation, we will firstly generate a t-dimension Boolean type column vector VC. We will change all of nonzero number according to the above rule in the ith row of the parent matrix if $VC_i = 1$, do nothing if $VC_i = 0$, and finally generate the new matrix. The constraints may not be satisfied that there may be the same nonzero number in the same column after row-based mutation operating. Therefore, we must run the above fixed program to make the new matrix satisfy the constraints.

(5) Column-Based Mutation. Similar to the row-based mutation, We will firstly generate a $T \times J$-dimension Boolean type row vector VC. We will change all of nonzero number according to the above rule in the jtth row of the parent matrix if $VC_{jt} = 1$, do nothing if $VC_{jt} = 0$, and finally generate the new matrix. Obviously, the new matrices also satisfy the constraints.

(6) Block-Based Mutation. Similar to the block-based crossover, block-based mutation is also operated for the unit of block corresponding to the period of t. Therefore, we will generate a T-dimension Boolean type row vector VC. We will change all of nonzero numbers according to the above rule in the tth block of the parent matrix if $VC_t = 1$, do nothing if $VC_t = 0$, and finally generate the new matrix. The constraints may not be satisfied that there may be the same nonzero number in the same column after block-based mutation operating. Therefore, we must run the above fixed program to make the new matrix satisfy the constraints.

3.4. Fitness Function and Selection Strategy. Based on the objective function, the fitness function will be given in this paper as follows:

$$F = \left(\alpha * \sum_{i=1}^{n} \left(\overline{\text{Profit}\,(Ta_i)} * x_i\right) + \beta * \sum_{i=1}^{n} x_i\right). \tag{24}$$

So as to select the new generation population, the penalty function for Ta_i will be given as follows:

$$pu_i = \begin{cases} 1, & \text{satisfied (18)} \\ \mu * \left(1 - \dfrac{T_i * x_i - T(P_i) * x_i}{\text{Max}\,(T_i)}\right), & \text{notsatisfied (18)}, \end{cases} \tag{25}$$

where μ denotes penalty factor. If the constraint (18) is satisfied in the task Ta_i, the penalty will not be carried out; that is to say the value of pu_i is 1. Otherwise, the penalty will be carried out, and if $x_i = 1$, the value of the $|T_i - T(P_i)|$ is bigger, and the penalty factor μ is also bigger. Therefore, we can give the penalty function for the feasible solution as follows:

$$Pu = \prod_{i=1}^{n} pu_i. \tag{26}$$

Based on the above function, the final fitness function in this paper is given as follows:

$$FF = F * Pu. \tag{27}$$

Apparently, we should reduce the number of the executed tasks or adjust the execution time or sequence of tasks if the constraint (18) is not satisfied in the feasible solution.

Traditional direct proportion selected strategy is adopted in this paper to generate offspring. In addition, we give the pressure factor for the fitness function so as to approach the optimal solution quickly.

For the individual s in the population whose size is *Size*, the fitness value is FF_s, and the selection probability is

$$\rho_s = \frac{FF_s - \chi * \text{Min}\,(FF)}{\sum_{s=1}^{Size} (FF_s - \chi * \text{Min}\,(FF))}, \tag{28}$$

where χ denotes pressure factor, and the value range is $[0, 1)$.

4. Experiments and Discussion

These experiments are performed based on the above example (described in Section 2.1) to test the effectiveness of the proposed method. We will perform the experiments with the conditions of enough resources and limited resources.

4.1. Experiments and Discussion with Enough Resources. We will evaluate the results affected by the correlation of resources with enough resources. The experiments are performed in the following 3 situations:

(i) the correlations are not taken into account E1;

(ii) only the correlations in a task are taken into account E2;

(iii) both the correlations in a task and among tasks are taken into account E3.

The parameters of the model are given as follows: the population size *Size* = 100, max generation is 100, the probability of crossover is 0.9, the probability of mutation is 0.05, the pressure factor $\chi = 0.9$, the penalty factor $\mu = 0.01$, and $\alpha = 0.5$ and $\beta = 0.5$. And the percent of correlation resources in a task is 30%; the percent of correlation resources among tasks is 40%.

The results of the experiments are shown in Figure 3.

In Figure 3, the y-coordinate denotes the total profits of all executed tasks. The x-coordinate denotes the number

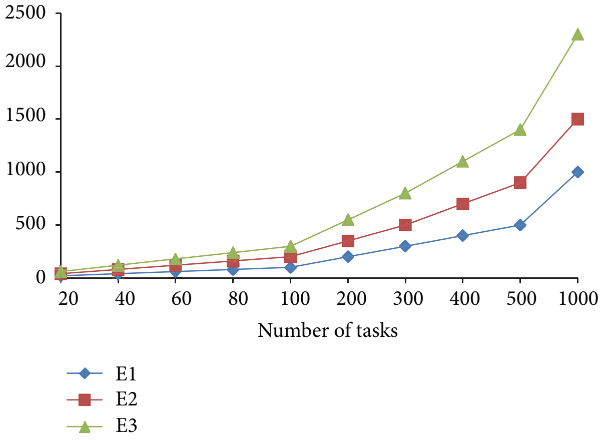

Figure 3: Results of the experiments with enough resources.

of tasks. With the number of tasks increased, the number of resources in corresponding candidate resources set is also increased to ensure that there are enough resources to execute all the tasks, but the percent of the correlation in a task and among tasks will not be changed.

As we can see in Figure 3, the total profit discrepancy in 3 situations is not bigger if the number of tasks is small (the number is less than 100). The reason is that the influence of the correlation of resources used in tasks is very small. But along with the increased number of tasks, the advantages of E2 and E3 which take correlations into account are huge, and the profits of E2 and E3 are much more than E1 for the same number of tasks, and the gap will be bigger and bigger. That is because the correlations both in a task and among tasks are considered, and the cost of selected resource will be lower than the default value. Therefore, it can be concluded that the proposed method that considered the correlations of resources has better performance than without considering the correlations.

4.2. Experiments and Discussion with Limited Resources. We will evaluate the results affected by the sharing of resources with limited resources. The experiments are performed in the following 2 situations:

(i) the sharing is not taken into account E4;

(ii) the sharing is taken into account E5.

The parameters of the model are given as follows: the population size *Size* = 200, max generation is 200, the probability of crossover is 0.95, the probability of mutation is 0.03, the pressure factor $\chi = 0.99$, the penalty factor $\mu = 0.01$, and $\alpha = 0.5$ and $\beta = 0.5$. And the percent of correlation resources in a task is 30%; the percent of correlation resources among tasks is 40%. The number of resources in each candidate resources set is fixed to 16, and the number of the given tasks varies from 2 to 30 which is an increment of 2.

The results of the experiments are shown in Figure 4.

In Figure 4, the y-coordinate denotes the total profits of the executed tasks. The x-coordinate denotes the number of given tasks. The number beside lines denotes the number of the executed tasks.

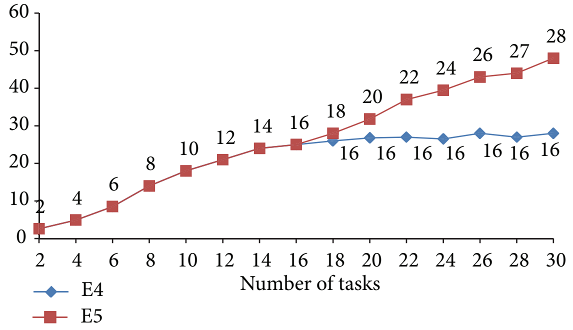

Figure 4: Results of the experiments with limited resources.

As we can see in Figure 4, the total profit and the number of executed tasks in E4 are the same as E5 if the number of tasks is less than 16. The reason is that all of the tasks could be executed without considering the sharing of resources. But along with the increased number of tasks, the number of tasks will not increase and keep 16 without considering the sharing of resources. But all the tasks can be executed with considering the sharing of resources until the number of given tasks is 28, and the profits are much more than without considering the sharing of resources. That is because the sharing of resources is considered. Therefore, it can be concluded that the proposed method that considered the sharing of resources has better performance than without considering the sharing and can solve the problem better for multitask oriented virtual resource integration and optimal scheduling which resources are limited.

5. Conclusions and Future Works

In this paper, the correlation and sharing of resources are both considered during multitask oriented virtual resources integration and optimal scheduling in cloud manufacturing, and the correlation and sharing models are presented. The problem is proposed and formulized, and GA based on the real number matrix code is employed to solve the problem. The experimental results show that the proposed methods are more effective and efficient for the problem.

Only the cost and time of Qos are considered in this paper; actually, there are many other factors to be considered such as reliability and trust. In the algorithm, there are a lot of invalid solutions in the solution space. In the future, we should avoid the invalid solutions during the process of algorithm or reduce the solution space by optimizing the code of solution to make the algorithm more effective. In addition, only one cloud manufacturing enterprise is considered; in the future, we should solve the problem for multienterprises.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This paper is supported by the National Science and Technology Support Program (Grant no. 2012BAF12B16), the 973 Program of China (Grant no. 2010CB328000), the National Natural Science Foundation of China (Grant nos. U1201251 and 61133016), and the National 863 Plan of China (Grant no. 2012AA040906).

References

[1] Z. de-Chen, C. Zhen, Z. Xi-Bin et al., "Manufacturing service and its maturity model," *Computer Integrated Manufacturing Systems*, vol. 18, no. 7, pp. 1584–1594, 2012.

[2] B.-H. Li, L. Zhang, S.-L. Wang et al., "Cloud manufacturing: a new service-oriented networked manufacturing model," *Computer Integrated Manufacturing Systems*, vol. 16, no. 1, pp. 1–7, 16, 2010.

[3] X. Xu, "From cloud computing to cloud manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 1, pp. 75–86, 2012.

[4] D.-C. Zhan, X.-B. Zhao, S.-Q. Wang et al., "Cloud manufacturing service platform for group enterprises oriented to manufacturing and management," *Computer Integrated Manufacturing Systems*, vol. 17, no. 3, pp. 487–494, 2011.

[5] C. S. Hu, C. D. Xu, X. B. Cao, and J. Fu, "Study of classification and modeling of virtual resources in Cloud Manufacturing," *Applied Mechanics and Materials*, vol. 121, pp. 2274–2280, 2012.

[6] K. Birman, G. Chockler, and R. van Renesse, "Toward a cloud computing research agenda," *ACM SIGACT News*, vol. 40, no. 2, pp. 68–80, 2009.

[7] R. Amorim, D. B. Claro, D. Lopes, P. Albers, and A. Andrade, "Improving web service discovery by a functional and structural approach," in *Proceedings of the 9th IEEE International Conference on Web Services (ICWS '11)*, pp. 411–418, Washington, DC, USA, July 2011.

[8] W. Tan, Y. Fan, M. Zhou, and Z. Tian, "Data-driven service composition in enterprise soa solutions: a petri net approach," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 3, pp. 686–694, 2010.

[9] W. Liu, B. Liu, D. Sun et al., "Study on multi-task oriented services composition and optimisation with the "Multi-composition for each task" pattern in cloud manufacturing systems," *International Journal of Computer Integrated Manufacturing*, vol. 26, no. 8, pp. 786–805, 2013.

[10] S. K. Bansal, A. Bansal, and M. B. Blake, "Trust-based dynamic web service composition using social network analysis," in *Proceedings of the IEEE International Workshop on Business Applications of Social Network Analysis (BASNA '10)*, pp. 1–8, Bangalore, India, December 2010.

[11] N. B. Mabrouk, S. Beauche, E. Kuznetsova et al., "QoS-aware service composition in dynamic service oriented environments," in *Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware*, p. 7, Springer, 2009.

[12] Y.-H. Shen and X.-H. Yang, "A self-optimizing QoS-aware service composition approach in a context sensitive environment," *Journal of Zhejiang University: Science C*, vol. 12, no. 3, pp. 221–238, 2011.

[13] Y. Wei and M. B. Blake, "Service-oriented computing and cloud computing: challenges and opportunities," *IEEE Internet Computing*, vol. 14, no. 6, pp. 72–75, 2010.

[14] F. Tao, D. Zhao, Y. Hu, and Z. Zhou, "Resource service composition and its optimal-selection based on particle swarm optimization in manufacturing grid system," *IEEE Transactions on Industrial Informatics*, vol. 4, no. 4, pp. 315–327, 2008.

[15] M. Alrifai and T. Risse, "Combining global optimization with local selection for efficient QoS-aware service composition," in *Proceedings of the 18th International Conference on World Wide Web*, pp. 881–890, ACM, 2009.