Int J Adv Manuf Technol (2016) 84:183–196
DOI 10.1007/s00170-015-7961-x

ORIGINAL ARTICLE

Service requirement conflict resolution based on ant colony optimization in group-enterprises-oriented cloud manufacturing

Xiaorong Huang[1] · Baigang Du[2] · Libo Sun[3] · Feng Chen[1] · Wei Dai[1]

Received: 19 May 2015 / Accepted: 11 October 2015 / Published online: 31 October 2015

Abstract Cloud manufacturing (CMfg) platform for group enterprises (GE) is a kind of private CMfg, which is to integrate and optimize GE's internal resources and capacity for large complex equipment manufacturing. The platform makes a connection between distributed and heterogeneous manufacturing resources to build a virtual pooling of resources for the group. However, it is evident that uncertainties and dynamics inherently exist in the platform and it causes manufacturing service requirement conflict among multiple projects. In order to address this issue, the management process of large complex equipment manufacturing project in GE-oriented CMfg platform is described. And next, this paper analyzes the causes and characteristic of service conflict in the platform. Then, a multi-objective mathematical model of conflict resolution is proposed. The proposed model considered both global target of shortest duration and partial target of tasks change minimization. Moreover, a method based on serial schedule generation scheme (SSGS) and ant colony optimization (ACO) algorithm is put forward to solve the model. Finally, a conflict resolution case study in a cement equipment manufacturing group enterprise is provided to illustrate the application of the proposed the model and algorithm.

✉ Xiaorong Huang
xiaorong.huang@vip.163.com

[1] School of Management and Economics, Hubei Polytechnic University, Huangshi, Hubei 435003, China

[2] School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China

[3] Sinoma Technology & Equipment Group Co., Ltd, Tianjing 300400, China

Keywords Conflict resolution · Cloud manufacturing (CMfg) · Group enterprises (GE) · Ant colony optimization (ACO) · Serial schedule generation scheme (SSGS)

1 Introduction

As a new service-oriented manufacturing mode, cloud manufacturing (CMfg) can handle more complex manufacturing issues and conduct on a larger scale collaborative manufacturing [1–4]. CMfg platform is divided into two: for small-medium enterprises (SME-oriented) and for group enterprises (GE-oriented) [5, 6]. GE-oriented CMfg platform, which is a kind of private CMfg for large complex equipment (LCE) manufacturing, focuses on supporting dynamic manufacturing resources sharing and collaboration in the group. In other words, the platform makes a connection between distributed and heterogeneous manufacturing resources to build a virtual pooling of manufacturing services for the group.

After receiving LCE orders, the group needs to develop manufacturing plan elaborately based on the characteristics of orders and manufacturing services in the cloud. Essentially, to serve multiple orders simultaneously, GE-oriented CMfg makes the manufacturing resources be better combined and achieve higher efficiency in a larger scale. However, due to the long duration of LCE manufacturing project, it is evident that uncertainties and dynamics inherently exist in GE-oriented CMfg [7], such as newly added tasks, task priorities change, and service failures. Even all projects in GE-oriented CMfg have been meticulously scheduled based with complex tasks interlinked through precedence rules, when the service provider cannot meet the demand, then there is a competitive requirement conflict that multiple tasks competing for the same manufacturing service. The conflict is a trigger for the disorder of GE-oriented CMfg management. Consequently, it causes the project delays and cost overruns.

There is an extensive literature concerning about cloud manufacturing, project scheduling, conflict prevention, and conflict resolution. However, in the context of GE-oriented CMfg, the problem of service requirement conflict among LCE manufacturing projects has not been given much attention in the literature, especially about conflict resolution in this environment. That is the motivation of this paper.

Here, we describe the management process of LCE manufacturing project in GE-oriented CMfg platform, especially about task assignment process. And next, the causes and characteristic of service requirement conflict in the platform are analyzed. This is the background and foundation of the paper. Drawing upon this, a multi-objective mathematical model of conflict resolution is proposed. Extant research about conflict resolution is mainly in the context of collaborative design, concurrent engineering, and other fields. The existing conflict resolution model is mainly rule based, case based, and negotiation based. Few studies have been carried out considering the problem with a global perspective and taking full account of previous plan. Without this, the ability of handling uncertainties and dynamics would still be limited. To improve this limit, the proposed novel model considered both global target of shortest duration and partial target of tasks change minimization.

The essence of solving the model is how to determine the priority of the tasks in the conflict. Changing the priority of current tasks may cause a conflict between subsequent tasks. Thus, if considering this problem with a holistic perspective, it would be more complex. With a global perspective, this problem can be considered as a resource-constrained scheduling problem (RCSP), a typical NP-hard problem. Researchers have developed a lot of intelligent algorithms to address it [8–11]. However, compared with the classical RCSP, Scheduling problem in this paper needs to be carried out on the basis of the original plan. It has not been substantially researched at this point. Thus, considering the above problems and situations, a novel intelligent algorithm namely ant colony optimization with serial schedule generation scheme (ACO-SSGS) is designed to solve the model. In this algorithm, ACO can constantly revise the priority of conflict task by using the volatilization and strengthening of pheromone.

The remainder of this paper is organized as follows. In the next section, we review the relevant literature related to cloud manufacturing service management and conflict settlement in the context of industrial project scheduling. Section 3 describes the management process of LCE manufacturing project in GE-oriented CMfg platform. Problem description and mathematic model of conflict resolution are developed in Section 4. Section 5 presents a method based on SSGS and ACO for solving the given model. A case study of conflict resolution for cement equipment manufacturing in a group enterprise is given in Section 6. Finally, conclusions are given in Section 7.

2 Literature review

2.1 Cloud manufacturing service management

In the past 6 years, CMfg has become a hot topic in the manufacturing industry [4, 12]. The concept [1, 2, 5, 13, 14], Key characteristics [7, 15], system architecture [5, 6, 13, 16], associated core support technologies [16–19], service management [7, 20–27], and the prototypes of CMfg and its application [3, 6, 20, 28, 29] have been widely studied. CMfg encapsulates manufacturing resources and manufacturing capabilities into manufacturing services [30], which can be managed and operated in an intelligent and unified way to enable a larger scale sharing and collaboration. Thus, service management of CMfg is a key issue to effectively improve the efficiency of sharing and collaboration [7]. The main research work in this area are as follows [31]: (1) service publishing, request, and discovery, which refer to service registration, publishing, searching, and matching; (2) quality of service management, which refer to service execution assurance, monitoring, and evaluation; (3) security, trust, and reliability management, which refer to identity check, authorization, access control, and user trust; (4) service composition and scheduling, which focuses on selecting the appropriate services and making the product scheduling according to task requirements and customer demands.

2.2 Conflict settlement

Many researchers have been focused on conflict settlement in the context of industrial project scheduling because it indicates great potential to improve the efficiency of resource utilization and to balance the total cost and the completion time of the projects. Huge number of publications on this issue can be found in the past decades. These publications can be classified into two categories, which are conflict prevention and conflict resolution.

2.2.1 Conflict prevention

There are considerable various uncertainties during project execution in real world. These uncertainties may stem from a number of possible sources [32, 33]: activities take more or less time than originally estimated, resources unavailable, unpredictable conditions, ill-defined and imprecise data which was used in scheduling, etc. Conflict prevention is a vital way to handle these uncertainties in advance. These methods can be classified into three categories, which are project buffer [34], fuzzy scheduling [35–38], and robust scheduling

[39–42]. (1) Project buffer mainly consist of time buffer and resource buffer, which deal with uncertainties by using safety time and resources redundancy for tasks and projects. (2) Resource requirement and time duration are often confronted with judgmental statements that are vague and imprecise. To deal with these vague and imprecise statements, fuzzy scheduling is widely used by using fuzzy number to form the fuzzy model. (3) Robust scheduling handles the uncertainties by fault tolerance, which can be achieved through resource redundancy or time redundancy.

2.2.2 Conflict resolution

Conflict prevention has been widely researched in issue of job scheduling, project scheduling, and resource configuring in CMfg. However, due to the complexity and dynamic of CMfg, it is very important to provide a timely and effective resolution scheme to the conflict when it occurred. Some researches have focused on this issue. To resolve resource conflict in the manufacturing grid (MGrid), Fei Tao [43] analyzed the causes of resource conflicts and put forward a solution based on event–condition–action (ECA) rules. Drawing upon ECA rules, resource conflict in a node of MGrid can be settled effectively and swiftly. However, to settle the conflict of LCE manufacturing projects, it is not only the current conflict node itself that we should consider but also with a global perspective in GE-oriented CMfg platform. Enlarging the scope of the problem will exponentially increase the complexity of the rules. Zhou and Wang [44] established resource capacity model which considers the resource as capability in CMfg project management. With the target of minimum global influence scope, the model removes the conflict by searching alternative resources which have the same capacity. This approach takes advantage of sharing resources in CMfg platform; the essence of this approach is resource utilization equalization in the cloud. But if the conflict was caused by insufficient resources (include alternative resources), some tasks and its successors should be postponed and replanned.

In addition, extant research about conflict resolution has also found in the context of collaborative design [45, 46], concurrent engineering [47], and large-scale project management [48]. The existing conflict resolution model is mainly rule based [46], case based [49] and negotiation based [48]. Rule-based strategy is to establish different conflict resolution rules for different types of conflict, such as mission-critical priority, the shortest waiting time, first come first served, and so on. Case-based strategy refers to utilize the specific case information available as historical precedence for proposing solutions to current conflict. Negotiation-based refers to find an agreement between two or more parties where participants exchange proposals which mainly by game-theoretic approaches and heuristic approaches.

LCE is a kind of complex product system with long manufacturing duration, complicated structures and parts, complex manufacturing processes, and large number of manufacturing participants [50]. Therefore, service requirement conflict caused by multiple LCE manufacturing projects in GE-oriented CMfg platform is more complex. The above approaches for our problem are still limited. For example, there is a considerable number of factors involved in CMfg conflict, each conflict is different, and it is difficult to establish case database; fixed rules cannot deal with complex issues.

3 Service requirement conflict in GE-oriented CMfg platform

3.1 Task assignment process

The subsidiaries of GE usually obtain LCE orders with individualized requirements by bidding. Manufacturing of such orders is always managed as a project, with interim products defined for each task of the project [35]. LCE has a complex structure and over 10,000 kinds of components. Many components are special products which need to design and manufacture to order; some components can only be processed on some specific equipment. Due to the complexity of the product, manufacturing resources in single enterprise cannot meet all needs of the order. In this case, for the unified management of the decentralized manufacturing resources which belong to the subsidiaries, GE needs to integrate them; GE also needs to integrate the needs of the various subsidiaries and its customers to achieve unified management of decentralized demand.

As shown in Fig. 1, GE-oriented CMfg platform, which is constructed by using the Group Intranet, integrates manufacturing resources of various subsidiaries within the group [5]. All kinds of manufacturing resources in CMfg are transformed into manufacturing capability by the platform. And then manufacturing capability are re-packaged and deployed as manufacturing service that can be rapidly provisioned and released [16]. By using manufacturing service which was provided by GE-oriented CMfg platform, GE's subsidiaries can work together to provide a product solution and get better synergy between its subsidiaries on product design, manufacturing, and sales service.

The matching process of complex equipment manufacturing task and manufacturing service in GE-oriented CMfg is illustrated in Fig. 2. The main steps of the process are as follows:

1. When the subsidiaries obtain the order, combined with scheduling orders and product BOM structure [51], project is decomposed to form a project task chain.

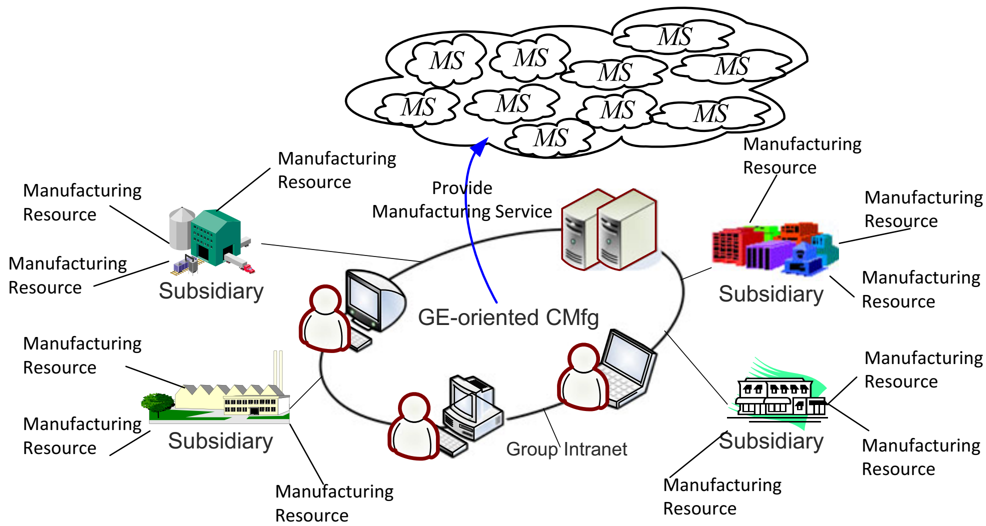


Fig. 1 GE-oriented CMfg platform

2. According to the characteristics of each task, manufacturing services in the cloud are clustered to form cloud servicers set which will be selected for the task by clustering algorithms [19].
3. Finally, the appropriate manufacturing service in each set which match up to the task is selected to form an optimal combination.

3.2 Manufacturing service requirement conflict

Generally, LCE should be manufactured according to the customers' specific requirements and the enterprise needs to make a unique engineering design or make changes from old products. The projects duration of products manufacturing are very long (about 1 or 2 years, or even longer); multiple projects are often implemented simultaneously in GE. Despite the different needs of various customers, the products in the same series have a certain similarity. Manufacturing tasks in different projects might require the same manufacturing service. In other words, the manufacturing service in the same cluster may be simultaneously required by different tasks. However, the manufacturing service capacity within a certain time is limited. Thus, to match, select, and combine cloud services, the capacity constraint of manufacturing service in GE-oriented CMfg must be taken into account inevitably.

The capacity of manufacturing services is dynamic in GE-oriented CMfg. This dynamic is reflected in the following

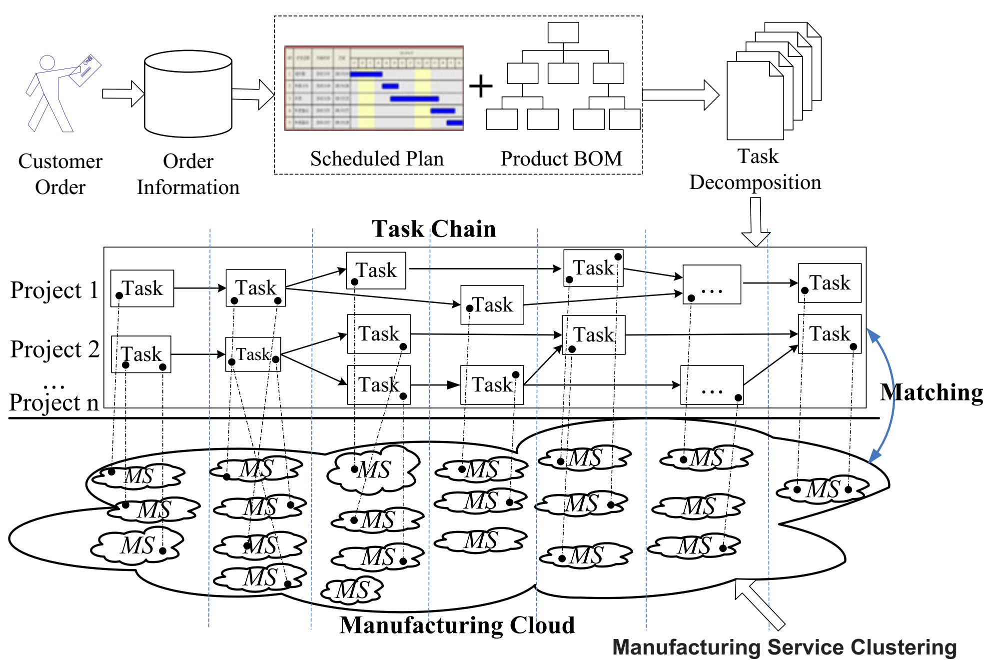


Fig. 2 Matching process of task and manufacturing service in GE-oriented CMfg

aspects: (1) Dynamic and uncertain manufacturing services demand. Before the project starts, task duration and its quantity of manufacturing services demanded are obscure and difficult to determine accurately. These uncertainties can easily affect the project scheduling [36]. (2) Dynamic and uncertain manufacturing resource in the cloud. External resources temporarily add to the manufacturing cloud, and some service capacities will be increased. Some services' failure or manufacturing resource leave from the manufacturing cloud, and some service capacities will be reduced. (3) Dynamic and uncertain external environment. Such as order cancel, rush orders insert and tasks priority change. Generally, these dynamics and uncertainty have been carefully considered when the project was planned. However, as a kind of complex production system, LCE is characterized by customized, interconnected complex subsystems, involves multiple collaborators, and has a long manufacturing period and easy to influence the effectiveness of its manufacturing project scheduling [52, 53]. In a certain period of time, for multiple tasks in different projects, if the demand on the same manufacturing cloud service is more than the supply, manufacturing service requirement conflict will arise. Manufacturing service requirement conflict is the priority conflict of multiple tasks using the same manufacturing service. Due to interconnected complex tasks, it is hard to determine which one should be delayed. When a task is delayed, it will affect its subsequent tasks and even other projects. This may cause the projects' duration and cost increases and make the original plan disorder.

In the next section, we will describe this issue and establish the mathematical model considering both global target of project duration and partial target of tasks change.

4 Problem description and modeling

4.1 Problem description

Drawing upon the above analysis, the problem can be described as follows: In GE-oriented CMfg platform, there are a number of LCE manufacturing projects when manufacturing service requirement conflict. These projects include a number of separate interrelated (i.e., precedence relationship) tasks. Each task with a fixed duration has planned start time and end time and requires corresponding manufacturing service. In a certain period, the limit of manufacturing service's capacity let some tasks compete for the same manufacturing service.

The problem is how to allocate limited manufacturing service to eliminate conflict between the tasks and to achieve optimal management objectives. Although this problem is very similar to the traditional resource-constrained project scheduling problem [32, 54], there are several significant differences between them: (1) All tasks have been scheduled reasonably. When eliminating conflict, the feasible solution is associated not only with the current task and its related services but also closely with the successor tasks. For example, eliminating the current conflict may lead to manufacturing service requirement conflict of its successor tasks. (2) Manufacturing service capabilities are dynamic and change with time. (3) The pros and cons of conflict resolution not only depend on the local impact of the previous projects plan but also take into account the global impact of all projects. In other words, eliminating the conflict requires to change the original plan as slightly as possible, simultaneously, to minimize the delay period for all projects.

4.2 Assumptions of the optimization model

Some assumptions have been made in order to strengthen modeling ability of the proposed model, as follows.

1. All projects are independent of each other.
2. There are precedence constraint relationship between various tasks in a project.
3. Within a certain period, the cloud service capabilities are known and has been fully gathered, a service exclusively for some specific categories of tasks.
4. Once a task is started, it cannot be interrupted; the required service is exclusive from other tasks until the end of the task.
5. When a conflict occurs, the tasks involved in the conflict and its subsequent tasks are not implemented; the conflict time is time 0.
6. The task execution time is fixed.

Before formulating a mathematical model of conflict resolution problem, the notations are defined as following that used throughout the remainder of this paper. The project set involved in the conflict is $P=\{p_1,p_2,\cdots,p_i,\cdots,p_N\}$; A_{ij} represents the j_{th} task that has not been executed in the project p_i; $T_{ij\_p\_s}$ represents the scheduled start time of task A_{ij}; $T_{ij\_a\_s}$ represents start time after conflict resolution of task A_{ij}; T_{ij_c} represents the duration of task A_{ij}; S_{ij_k} represents the required capabilities of cloud service s_k for finishing task A_{ij}, k is the category code of cloud service, $k=1,2,\cdots,K$; $S\_foruse(t,s_k)$ represents how many capabilities that cloud service s_k can supply at time t; I_t represents tasks set in the execution state at time t; N represents the number of project in conflict; J_i represents how many tasks has not been executed in the project i in conflict; X represents how many tasks' previously scheme are changed after conflict resolution; $Y(p_i)$ represents the delay time of the project p_i after conflict resolution.

4.3 Conflict resolution mathematical model

The conflict resolution mathematical model is given as follows:

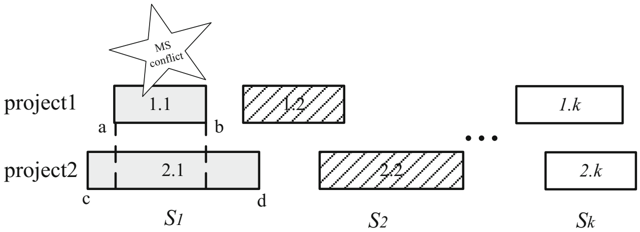


Fig. 3 Conflict resolution process

$$\min X=\sum_{i=1}^{N}\sum_{j=1}^{J}x_{ij},x_{ij}=\begin{cases}1, & \text{if } \left(T_{ij\_a\_s}\neq T_{ij\_p\_s}\right)\\ 0, & \text{else}\end{cases} \tag{1}$$

$$\min\sum_{i=1}^{N}Y(p_i) \tag{2}$$

$$T_{ij\_a\_s}\geq\left(T_{ih\_a\_s}+T_{ih\_c}\right),\left(\forall i,j,\forall(i,h)\in P_{ij}\right) \tag{3}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{J_i}S_{ij\_k}\leq S\_\text{foruse}\ (t,s_k),\left((i,j)\in I_t,k=1,2,\cdots,K\right) \tag{4}$$

$$Y(p_i)=\max\left\{0,\max\left(T_{ij\_a\_s}\right)-\max\left(T_{ij\_p\_s}\right)\right\},(j=1,2,\cdots,J_i) \tag{5}$$

$$\min\left(T_{ij\_a\_s}\right)\geq\min\left(T_{ij\_p\_s}\right) \tag{6}$$

The objective function (1) minimizes the total number of tasks that this previous scheme are changed, and objective function (2) minimizes the total delay time of all projects. Function (3) represents the constraint of tasks precedence relationship within each project. Function (4) represents the constraint of service capacity in each time. Function (5) represents the delay time of the project p_i. Function (6) represents the constraint that the earliest start of each task cannot be earlier than previous scheme.

5 Solution approach

5.1 Handling for multi-objective optimization problem

A large amount of literature has provided handling methods for multi-objective optimization problem. Generally, most of them can be classified into three categories [55]: (1) aggregate objective function-based methods. (2) Pareto-compliant ranking-based methods. (3) Constrained objective function. Two objectives in the model we proposed are contradictory in some instances, but some instances are not. In other words, for conflict resolution, to minimize the total number of tasks that its plan needs to be changed may enlarge the delay time of all projects. But in some cases, to minimize the first objective will also minimize the second one. So, in this paper, we adopt weighted sum method to convert the target. Due to the different dimension of two targets, we need to standardize these two objectives.

In sum, the aggregate objective function has the following form.

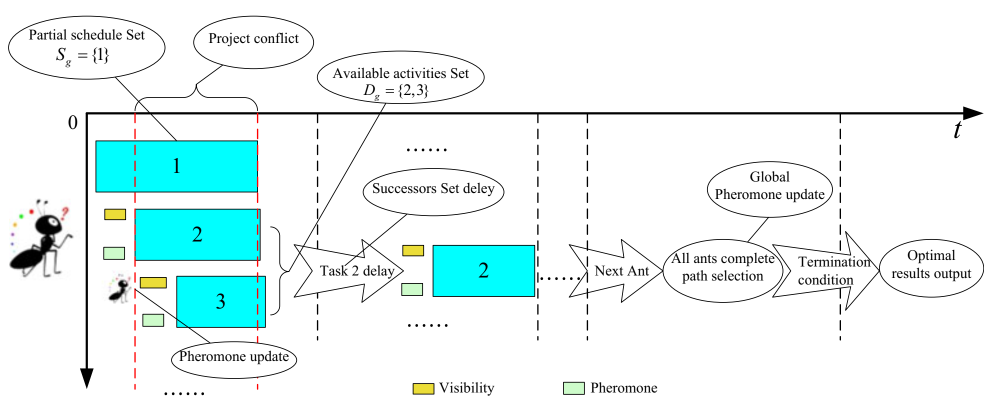
$$\min\left(\omega_1\times\frac{X}{\sum_{i=1}^{N}J_i}+\omega_2\times\sum_{i=1}^{N}\frac{Y(p_i)}{T_{iJ_i\_p\_s}}\right) \tag{8}$$

ω_1,ω_2 denote two different weights of the objectives, respectively.

5.2 Solution space

Different from RCPSP, solution space size of this model has a close relationship with the influence that the results of the current conflict resolution exert on its successor. The example is as shown in Fig. 3; there are two scheduled projects in the cloud. When conflict occurs, each project have k tasks with k kinds of cloud services. In the period ab, the supply of service S1 cannot afford the needs of tasks 1.1 and 2.1. There are two ways to eliminate the conflict: postpone task 1.1 or task 2.1 (the successors should be postponed) to satisfy $t_d\leq t_a$ or $t_b\leq t_c$. If there is no impact on their successors, solution space size will be two. In addition, there is an extreme case that the elimination of each conflict will lead to the conflict of its followed tasks. In this case, solution space size is 2^k. Therefore, the size of the solution space will vary within $2\sim 2^k$. Moreover, with the increase in the

Fig. 4 ACO-SSGS for conflict resolution



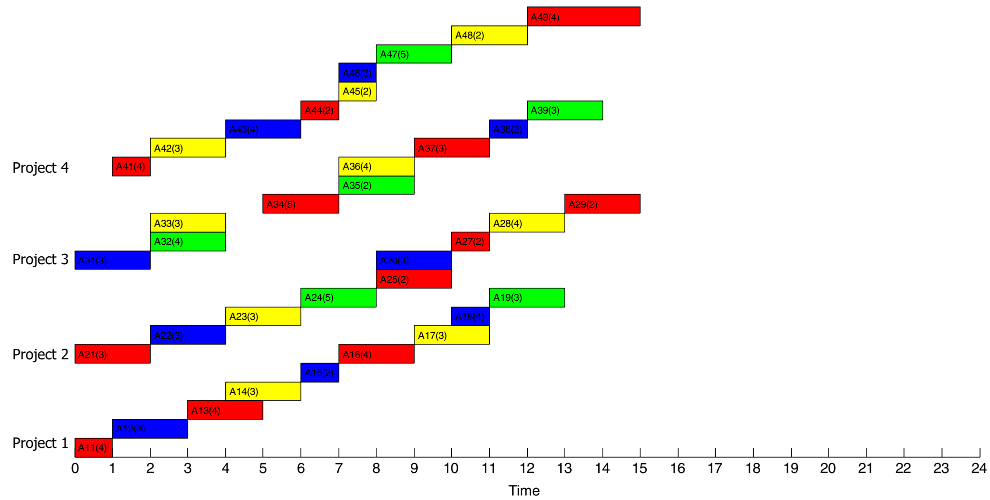


Fig. 5 The project Gantt chart in the private cloud at time 0

number of projects involved in the conflict, its solution space will vary over a larger area.

5.3 ACO-SSGS for conflict resolution

A large body of literature has provided support for RCPSP, which can be divided into deterministic algorithm and heuristic algorithm [56]. Deterministic algorithm can get the optimal solution but with the expansion of the scale of the problems that may have a "combinatorial explosion" phenomenon. It leads to low computational efficiency and even impossible to obtain solutions. Fortunately, heuristic algorithm can overcome this shortcoming; it mainly includes priority rule-based algorithms and intelligent algorithms. Priority rule-based algorithms are generally constituted by schedule generation scheme (SGS) and priority rule (PR). The purpose of SGS is to produce a feasible schedule plan, usually from a partial schedule, and gradually expanded until it generates a complete schedule. PR is used to determine the priority order of the tasks, that is, different tasks given a certain weight. This algorithm is simple, suitable, and has high computational efficiency; however, PR has poor directionality, lack universality, and limited search capability. In recent years, intelligent algorithms play an important role to address this problem, such as GA [57, 58], particle swarm optimization (PSO) [21], ACO [59], and so on. For GA, due to the change of solution space and the constraints of precedence and resources, the chromosomes are prone to illegal in evolution. GAs and PSO also lack in PR feedback.

Although applying the algorithm in cloud environment and traditional environment has some similarities, there are some significant differences. First, resources are dynamic and changing in cloud, but in traditional environment, they are fixed. Second, the algorithm needs to develop specific scheduling policy for a specific application in traditional environment; in cloud, task scheduling is generally not limited to specific application, which can support many types of applications, and can run multiple

Table 1 The projects data in conflict time

Task no.	ST	ET	ST	SR	Task no.	ST	ET	ST	SR	Task no.	ST	ET	ST	SR
A11	0	1	1	4	A24	6	8	2	5	A37	9	11	1	3
A12	1	3	3	3	A25	8	10	1	2	A38	11	12	3	2
A13	3	5	1	4	A26	8	10	3	3	A39	12	14	2	3
A14	4	6	4	3	A27	10	11	1	2	A41	1	2	1	4
A15	6	7	3	2	A28	11	13	4	4	A42	2	4	4	3
A16	7	9	1	4	A29	13	15	1	2	A43	4	6	3	4
A17	9	11	4	3	A31	0	2	3	3	A44	6	7	1	2
A18	10	11	3	4	A32	2	4	2	4	A45	7	8	4	2
A19	11	13	2	3	A33	2	4	4	3	A46	7	8	3	3
A21	0	2	1	3	A34	5	7	1	5	A47	8	10	2	5
A22	2	4	3	3	A35	7	9	2	2	A48	10	12	4	2
A23	4	6	4	3	A36	7	9	4	4	A49	12	15	1	4

Table 2 Service supply at each time

Time	S1	S2	S3	S4	Time	S1	S2	S3	S4	Time	S1	S2	S3	S4
0	6	3	5	2	8	6	7	6	4	16	4	5	4	4
1	6	3	5	2	9	5	5	5	4	17	4	5	4	4
2	4	4	5	6	10	5	3	5	5	18	4	5	4	4
3	4	4	6	6	11	2	3	5	6	19	6	6	6	6
4	4	4	6	6	12	4	6	4	4	20	6	6	6	6
5	5	4	6	6	13	6	5	4	4	21	6	6	6	6
6	7	6	6	2	14	6	5	4	4	22	6	6	6	6
7	4	7	6	6	15	4	5	4	4	23	6	6	6	6

Table 3 ACO parameter settings

Parameter	Denotation	Setting
Pheromone emphasis factor	α	0.2
Visibility emphasis factor	β	4
Volatile coefficient	ρ	0.1
Iteration number	NC_max	100
The initial concentration of pheromone	$\tau_{tA_{ij}}$	1
Weights of the objectives	ω_1,ω_2	0.5,0.5

applications simultaneously. Finally, the target is relatively simple in traditional environment, but in cloud environment, its scheduling strategy should be possible not only to improve the cloud service provider's service revenue but must also try to meet a large number of users for different resource types application requirements.

To fill the above gaps, in this paper, we propose an ACO combined with and SSGS [37, 60] for solving the model. ACO can change the priority dynamically by using the volatilization and strengthening of pheromone.

5.3.1 The design of ACO-SSGS

SSGS contains $J+1$ stages, each stage $g(g=0,1,2,\cdots,J)$ has partial scheduling scheme set S_g and available tasks set D_g for each stage. Tasks that already scheduled are contained in S_g. D_g contains the conflicting tasks in the current stage. It is evident that the predecessors of all task in D_g have contained in S_g. In each stage, we can get the state transition rules by pheromones and visibility of ants; then according to the rules, one task is selected from D_g and other tasks are postponed. By constantly updating pheromone, the optimal solution can be found eventually, as shown in Fig. 4.

Next, we will give a detailed description about state transition rule, pheromone update mechanism and algorithm process.

5.3.2 State transition rule

Ants crawling do not constitute a loop, so a virtual node as the start node ants is added. All ants are from this node to traverse all other nodes. The probability of the k_{th} ant choose task A_{ij} at time t according to the following formula.

$$P^k_{tA_{ij}} = \begin{cases} \dfrac{\tau^{\alpha}_{tA_{ij}} \bullet \eta^{\beta}_{tA_{ij}}}{\sum\limits_{t\in T} \tau^{\alpha}_{tA_{ij}} \bullet \eta^{\beta}_{tA_{ij}}}, A_{ij} < A_{\max} \\ 0, \text{otherwise} \end{cases}$$

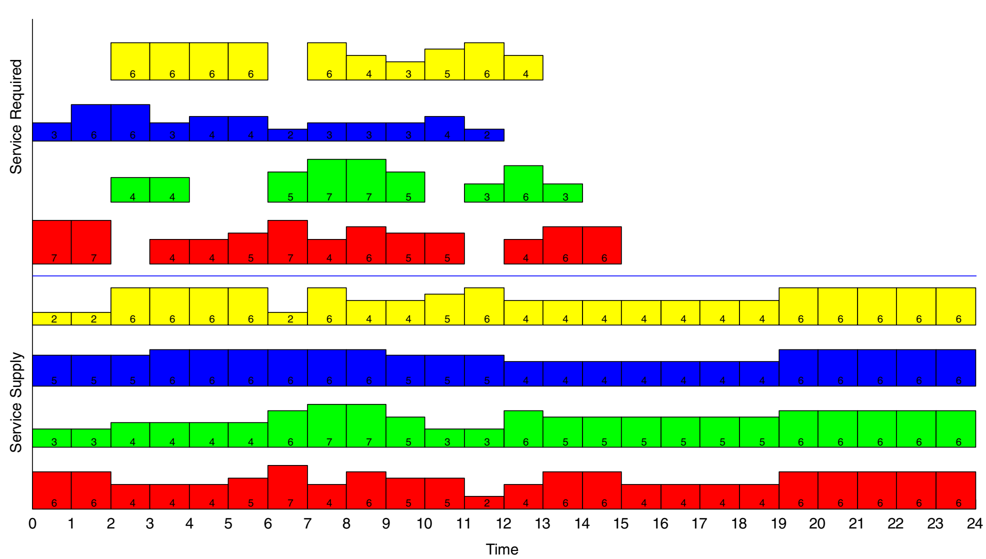


Fig. 6 The required capabilities of service and its supply at each time

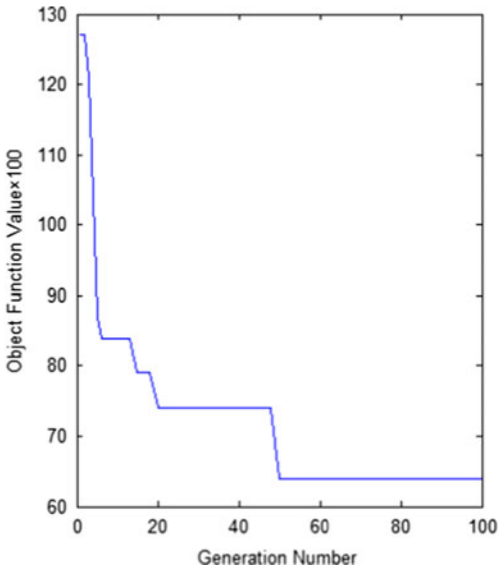


Fig. 7 The convergence curve of ACO-SSGS for conflict resolution

where $\tau_{tA_{ij}}$ is the pheromone of task A_{ij} at time t, $\eta_{tA_{ij}}$ is the visibility of task A_{ij} at time t, $A_{\max}$ is the maximum amount of tasks in the conflict set, and α and β represent the emphasis degree of visibility and pheromone, respectively. $\eta_{tA_{ij}}$ is calculated by Eq. (10):

$$\eta_{tA_{ij}} = \frac{1}{T_{ij\_c}} + \frac{1}{S_{ij\_k}} \tag{10}$$

where visibility is mainly determined by the duration and service requirement of the task.

5.3.3 Pheromone update mechanism

Drawing upon the rationale of SSGS, the proposed algorithm is mainly used the way of time scanning for conflict detection and resolution. On the side (t, A_{ij}), the unit length pheromone left by the k_{th} ant is determined by the traveled path of the ant. v_k is the total path length traveled by the k_{th} ant. The pheromone increment of unit length can be calculated using the following equation.

$$\Delta\tau_{tA_{ij}}^{k} = \begin{cases} \frac{Q}{v_k}, t, A_{ij} & \in S^* \\ 0, t, A_{ij} & \notin S^* \end{cases}$$

where $\Delta\tau_{tA_{ij}}$ is the pheromone increment of task A_{ij} at time t, Q is constant, and the updated pheromone can be calculated using the following equation.

$$\tau_{tA_{ij}} = (1-\rho)\tau_{tA_{ij}} + \Delta\tau_{tA_{ij}}$$

In the equation, ρ is the volatile coefficient.

5.3.4 Algorithm process

The detailed solving process of the proposed algorithm is as follows.

Step 1. Data initialization. The initialized data needs to be divided into two categories. One is algorithm parameters, including ant pheromone matrix, pheromone volatile coefficient, the maximum number of iterations, the initial iterations, maximum scan time. Another one is the data of tasks and cloud services, such as tasks code, tasks scheduled start time, task duration, the capabilities of cloud services, and task required capabilities of cloud services.

Step 2. Iterative algorithm initialization. The current time g is 0, the current of cloud service number k is 1. Current ant m start to look for the path.

Step 3. Scan the current task set. The current task set refers to all tasks which have not been implemented in

Fig. 8 The Gantt chart of conflict resolution result

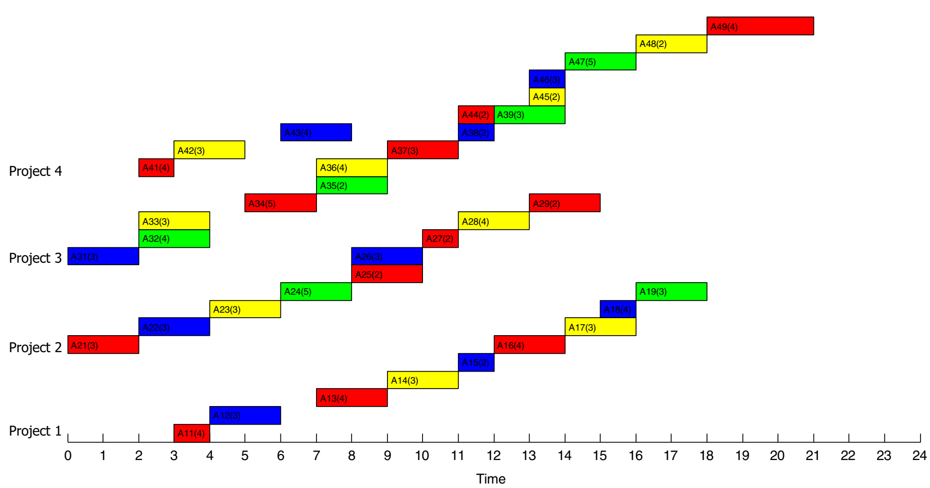


Table 4 The start time of each task after conflict resolution

Task no.	Start time	Task no.	Start time	Task no.	Start time	Task no.	Start time
A11	3	A21	0	A31	0	A41	2
A12	7	A22	2	A32	2	A42	3
A13	12	A23	4	A33	2	A43	5
A14	13	A24	6	A34	5	A44	7
A15	15	A25	8	A35	7	A45	9
A16	16	A26	8	A36	7	A46	10
A17	18	A27	10	A37	9	A47	14
A18	19	A28	11	A38	11	A48	16
A19	20	A29	13	A39	12	A49	18

the cloud at current time g. Put the tasks that required cloud service number k at current time g into D_g^k.

Step 4. Conflict judgment. If there is no conflict in D_g^k, go to step 9, otherwise, turn to the next step.

Step 5. Get the conflict duration t_during by scanning D_g^k, where $g \le t \le t_{\max}$, $t_{\max}$ is the earliest completion time of the tasks in D_g^k. Searching for the tasks that included during the time interval t_during and put them into D_g^k.

Step 6. Update D_g^k. (1) Remove the tasks that have been planned by the proposed algorithm from D_g^k. $D_g^k = D_g^k - D_g^k \cap S_g$. (2) If $S_{ij\_k} \ge S\_\text{foruse}(t, s_k)$, Remove task A_{ij} from D_g^k and put it into delayed set Delay_g^k. At the same time, update $S\_\text{foruse}(t, s_k)$, $g \le t \le t_{\max}$.

Step 7. If D_g^k is empty, turn to the next step. Otherwise, select the tasks in D_g^k by using ACO, simultaneously put the selected tasks into S_g and selecting set Choose_g. Update $S\_\text{foruse}(t, s_k)$, $g \le t \le t_{\max}$.

Step 8. Delay all tasks in Delay_g^k and its successors. The delay time of the tasks belong to project p_i is Δt_i, $\Delta t_i = t_{i\_\min} - T_{ij\_a\_s}$, where $t_{i\_\min}$ is the minimum completion time of all tasks in Choose_g.

Step 9. Judge whether all cloud services traverse completed at time g. If not, cloud service $k=k+1$, turn to step 3. Otherwise, turn to the next step.

Step 10. Judge whether all time traverse completed. If not, scanning time $g=g+1$. Otherwise, turn to the next step.

Step 11. Record resolution scheme of the current ant and judge whether this ant is the last. If not, ant $m=m+1$, turn to step 2. Otherwise, turn to the next step.

Step 12. Pheromone volatilization and strengthening. Volatilize the pheromones of all paths. Strengthen the pheromone of the shortest path. Iteration number $NC = NC+1$.

Table 5 Algorithm parameters of GA

PopSize	*MaxGen*	P_c	P_m
100	100	0.8	0.05

Step 13. Terminate judgment. If the termination condition is satisfied, terminate the procedure and output the results; otherwise turn to step 2.

6 Illustrative example

6.1 The example data

In order to demonstrate the application of the proposed model and algorithm, we give a conflict resolution case study in a cement equipment manufacturing group enterprise. The products in the case group enterprise are large complex cement equipment for cement production line, such as cement burning equipment, grinding equipment, and crusher. The group has nearly 23 subsidiaries mainly distributed in Jiangsu and Hebei provinces in China. To share manufacturing resources, idle devices, and raw materials of each member, the group has built the prototype system of GE-oriented CMfg platform.

On the prototype system, we simulate a service requirement conflict, and then project conflict set is obtained. In the set, the project Gantt chart is shown in Fig. 5. There are four projects in the set; tasks are represented by a rectangle, time 0 is the current time. In the rectangle, different colors represent different service requirements (i.e., the red represent the requirement of cloud service 1), the characters and numbers represent task code and the required capabilities (i.e., A41(4) represent task A41 and its required capabilities is 4). The project data is shown in Table 1, and Service supply at each time is shown in Table 2.

The required capabilities of service and its supply at each time are shown in Fig. 6. From Figs. 5 and 6, it is distinct that A11 compete with A12 on service 1 at time 0; A12, A22, and A31 compete against each other on service 2 at time 1 and time 2.

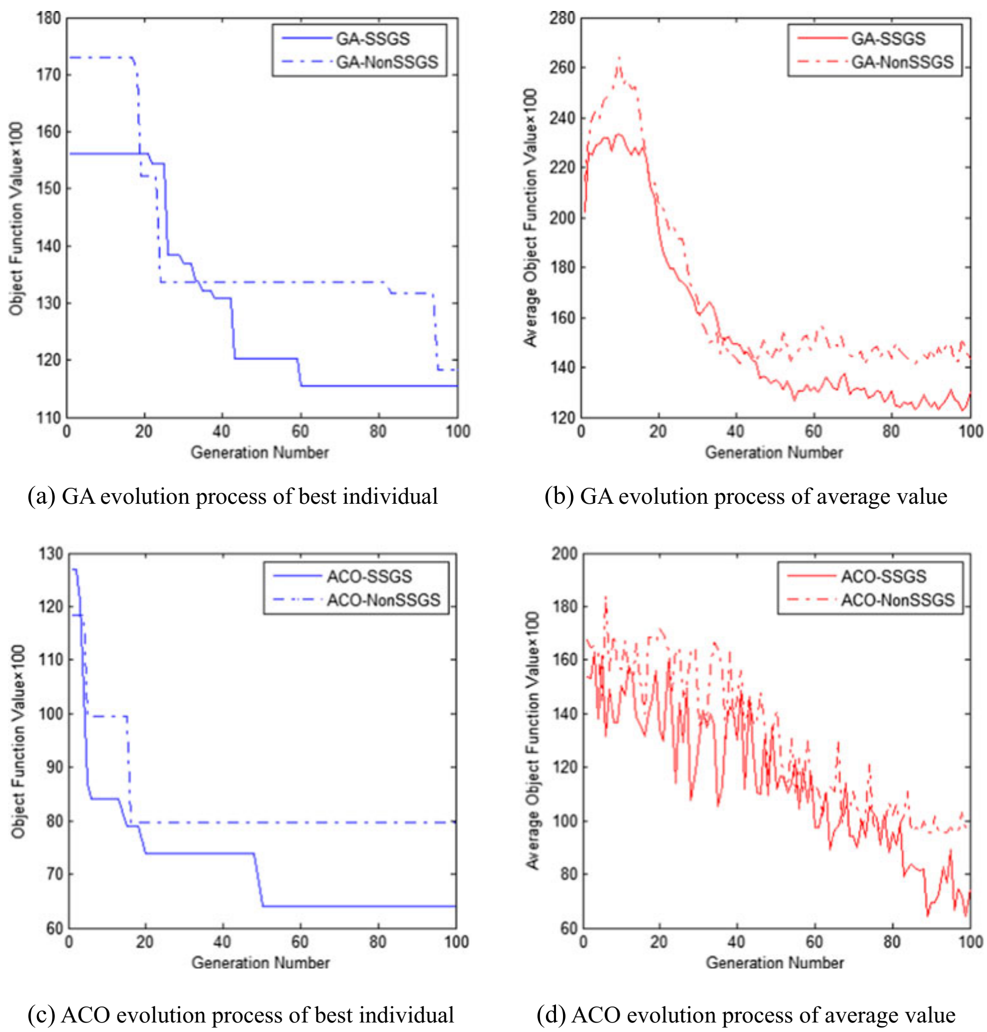
Table 6 The results of comparison experiment

	Initial best fitness	Converged best fitness	Elapsed time/s
GA-SSGS	(1)150.34 (2)160.96 (3)147.78 (4)156.2 (5)165.62	(1)134.47 (2)133.59 (3)131.68 (4)115.5 (5)150.85	All more than 500
ACO-SSGS	(1)116.41 (2)105.41 (3)147.78 (4)127.22 (5)152.33	(1)64.23 (2)79.62 (3)77.31 (4)74.51 (5)64.23	(1)52.31 (2)51.83 (3)52.05 (4)50.76 (5)52.65
GA-NonSSGS	(1)173 (2)105.41 (3)167.8 (4)168.2 (5)157.2	(1)118.4 (2)157.7 (3)127.6 (4)147.5 (5)148.4	All more than 500
ACO-NonSSGS	(1)118.3 (2)152.8 (3)124.4 (4)116.1 (5)131.9	(1)79.62 (2)95.62 (3)95.62 (4)79.62 (5)85.02	(1)54.36 (2)52.79 (3)52.80 (4)51.49 (5)53.53

6.2 Computational outcomes

The computational software and hardware platform parameters are listed as follows: Windows 8.1, MATLAB R2014a, Intel(R) Core(TM) i7-4510U CPU, 2.0GHz, and 16G RAM. In order to use the algorithm in Cmfg platform, the MATLAB codes are compiled into a DLL file and embed into the platform. The processes are carried out with three steps.

Step 1. Use deployment tool in MATLAB to compile MATLAB functions into DLL file with class name *GAClass*.



(a) GA evolution process of best individual

(b) GA evolution process of average value

(c) ACO evolution process of best individual

(d) ACO evolution process of average value

Fig. 9 Evolution process of GA and ACO. **a** GA evolution process of best individual. **b** GA evolution process of average value. **c** ACO evolution process of best individual. **d** ACO evolution process of average value

Step 2. Add reference of compiled DLL file into Cmfg platform.

Step 3. Call the main function *mainFUN* of the proposed algorithm. The process to call *mainFUN* of *GAClass* is shown in the algorithm 1.

Algorithm 1: Algorithmic flow for calling *mainFUN* in Cmfg platform

```
//initialize parameter
MWNumericArray TaskNumber= Convert.ToInt32(txtTaskNumber.Text.Trim());
MWNumericArray TimeMax= ……
MWNumericArray PopSize =Convert.ToInt32(txtPopSize.Text.Trim());
MWNumericArray Pc = ……
GAClass ga = new GAClass();
MWArray[] agrsOut = new MWArray[2];
MWArray[] agrsIn = new MWArray[] {TaskNumber, TimeMax, … , PopSize, Pc, Pm, LeagueSize, MaxGen};
ga.mainFUN(2, ref agrsOut,agrsIn);
MWNumericArray ResultBestInd = agrsOut[0] as MWNumericArray;
MWNumericArray ResultBestFitness = agrsOut[1] as MWNumericArray;
Return ResultBestInd and ResultBestFitness
```

ACO parameter settings are shown in the following Table 3. Figure 7 is the convergence curve of ACO-SSGS for the example, including the convergence curve of the value and the average value of objective function. It is evident that the proposed algorithm converges around 50 iterations. The Gantt chart of conflict resolution result is illustrated in Fig. 8, and the start time of each task is shown in Table 4.

Comparing the result with the projects scheduling before the conflict, projects 1 and 4 are delayed to eliminate all conflicts, the total delay time is 12, some task (i.e., A41, A42) are changed slightly. Considering the services' requirement and its supply, the result achieves the intended purposes.

6.3 Comparison experiment

To demonstrate the effectiveness of ACO-SSGS, the comparison experiment among GA-SSGS, ACO-SSGS, GA without SSGS strategy (GA-NonSSGS) and ACO without SSGS strategy (ACO-NonSSGS) have been carried out. As a kind of Heuristic algorithm, GA provides a common framework for solving complex system optimization problems [61], it does not depend on the field of specific problem and has strong robustness on various issues. Therefore, GA is widely used in many fields [62], such as function optimization, combinatorial optimization, production scheduling, automatic control, project scheduling and so on.

Table 5 gives the algorithm parameters of GA. All the algorithms are running for five times and the results are shown in Table 6. It is easily observed that ACO-SSGS has advantages in solving efficiency and quality. For a more straightforward description, the best individuals and average objective function values are employed as metric to illustrate the evolution process shown in Fig. 9. Comparing the evolution process of GA and ACO, it is obvious that the obtained result by ACO is better than GA with or without SSGS strategy. However, comparing the evolution process of GA and ACO of SSGS and without SSGS strategy, graphs (a) and (c) in Fig. 9 demonstrated that the SSGS strategy was more efficient for optimal solution searching. The advantage is benefited from the SSGS strategy, which a Tabu search collection is used to decrease the searching space and avoid partial optimal solution.

7 Conclusions

In this research, we have made an attempt to resolve the conflict resolution problem in GE-oriented CMfg for LCE manufacturing. We have described the management process of LCE manufacturing project and analyzed service conflict in the platform. Considering the characteristics of the conflict, we have proposed a multi-objective mathematical model considered both global target of shortest duration and partial target of tasks change minimization. To solve the model, the algorithm based on SSGS and ACO has been put forward. ACO can change the priority of tasks dynamically according to the feedback of the ants. In this way, the conflict can be well settled.

There are a number of opportunities to expand the proposed research. In this research, we used the delay time and the number of the changed tasks as objective; all time are rigid. Concerning these limitations, the suggestions for further research are as follows: other objectives can be considered such as delay cost, multi-mode of the task duration with different costs, and the fuzziness of task time.

Acknowledgments This study has been financially supported by the Humanities and Social Sciences Youth Fund Project of Ministry of Education of China (No. 13YJCZH028), National Natural Science Foundation of China (No. 71473074), and Hubei Provincial Department of Education (No. Q20144403)

References

1. Li B-H, Zhang L, Wang S-L, Tao F, Cao J, Jiang X, Song X, Chai X (2010) Cloud manufacturing: a new service-oriented networked manufacturing model. Comput Integr Manuf Syst 16(1):1–7

2. Zhang L, Luo Y, Tao F, Li BH, Ren L, Zhang X, Guo H, Cheng Y, Hu A, Liu Y (2012) Cloud manufacturing: a new manufacturing paradigm. Enterp Inf Syst 8(2):167–187. doi:10.1080/17517575.2012.683812
3. Wu D, Rosen DW, Wang L, Schaefer D (2015) Cloud-based design and manufacturing: a new paradigm in digital manufacturing and design innovation. Comput Aided Des 59:1–14. doi:10.1016/j.cad.2014.07.006
4. Tao F, Cheng Y, Zhang L, Nee AYC (2015) Advanced manufacturing systems: socialization characteristics and trends. J Intell Manuf: 1–16. doi:10.1007/s10845-015-1042-8
5. Li B-H, Zhang L, Ren L, Chai X-D, Tao F, Luo Y-L, Wang Y-Z, Yin C, Huang G, Zhao X (2011) Further discussion on cloud manufacturing. Comput Integr Manuf Syst 17(3):449–457
6. Huang B, Li C, Yin C, Zhao X (2013) Cloud manufacturing service platform for small- and medium-sized enterprises. Int J Adv Manuf Technol 65(9–12):1261–1272. doi:10.1007/s00170-012-4255-4
7. Tao F, Zhang L, Guo H, Luo Y-L, Ren L (2011) Typical characteristics of cloud manufacturing and several key issues of cloud service composition. Comput Integr Manuf Syst 17(3):477–486
8. Khalili S, Najafi AA, Niaki STA (2013) Bi-objective resource constrained project scheduling problem with makespan and net present value criteria: two meta-heuristic algorithms. Int J Adv Manuf Technol 69(1–4):617–626. doi:10.1007/s00170-013-5057-z
9. Bagherinejad J, Majd ZR (2014) Solving the MRCPSP/max with the objective of minimizing tardiness/earliness cost of activities with double genetic algorithms. Int J Adv Manuf Technol 70(1–4):573–582. doi:10.1007/s00170-013-5303-4
10. Qiong J, Yoonho S (2013) An improved particle swarm optimization for the resource-constrained project scheduling problem. Int J Adv Manuf Technol 67(9–12):2627–2638. doi:10.1007/s00170-012-4679-x
11. Yoosefzadeh HR, Tareghian HR (2013) Hybrid solution method for resource-constrained project scheduling problem using a new schedule generator. Int J Adv Manuf Technol 66(5–8):1171–1180. doi:10.1007/s00170-012-4398-3
12. Li J, Tao F, Cheng Y, Zhao L (2015) Big Data in product lifecycle management. Int J Adv Manuf Technol: 1–18. doi:10.1007/s00170-015-7151-x
13. Tao F, Zhang L, Venkatesh VC, Luo Y, Cheng Y (2011) Cloud manufacturing: a computing and service-oriented manufacturing model. Proc Inst Mech Eng B J Eng Manuf 225(10):1969–1976
14. Xu X (2012) From cloud computing to cloud manufacturing. Robot Comput Integr Manuf 28(1):75–86. doi:10.1016/j.rcim.2011.07.002
15. Wu D, Rosen DW, Wang L, Schaefer D (2014) Cloud-based manufacturing: old wine in new bottles? Procedia CIRP 17:94–99. doi:10.1016/j.procir.2014.01.035
16. Vincent Wang X, Xu XW (2013) An interoperable solution for cloud manufacturing. Robot Comput Integr Manuf 29(4):232–247. doi:10.1016/j.rcim.2013.01.005
17. Tao F, Zuo Y, Xu L, Zhang L (2014) IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing. IEEE Trans Ind Inform 10(2):1547–1557. doi:10.1109/TII.2014.2306397
18. Tao F, Cheng Y, Xu L, Zhang L, Li B (2014) CCIoT-CMfg: cloud computing and internet of things-based cloud manufacturing service system. IEEE Trans Ind Inform 10(2):1435–1442. doi:10.1109/TII.2014.2306383
19. Liu N, Li X, Shen W (2014) Multi-granularity resource virtualization and sharing strategies in cloud manufacturing. J Netw Comput Appl 46:72–82. doi:10.1016/j.jnca.2014.08.007
20. Cheng Y, Tao F, Liu Y, Zhao D, Zhang L, Xu L (2013) Energy-aware resource service scheduling based on utility evaluation in cloud manufacturing system. Proc Inst Mech Eng B J Eng Manuf 227(12):1900–1914. doi:10.1177/0954405413492966
21. Tao F, Zhao D, Hu Y, Zhou Z (2008) Resource service composition and its optimal-selection based on particle swarm optimization in manufacturing grid system. IEEE Trans Ind Inform 4(4):315–327. doi:10.1109/TII.2008.2009533
22. Tao F, Laili YJ, Xu L, Zhang L (2013) FC-PACO-RM: a parallel method for service composition optimal-selection in cloud manufacturing system. IEEE Trans Ind Inform 9(4):2023–2033. doi:10.1109/TII.2012.2232936
23. Tao F, Cheng Y, Zhang L, Zhao D (2012) Utility modelling, equilibrium, and coordination of resource service transaction in service-oriented manufacturing system. Proc Inst Mech Eng B J Eng Manuf 226(6):1099–1117. doi:10.1177/0954405412438011
24. Tao F, Guo H, Zhang L, Cheng Y (2012) Modelling of combinable relationship-based composition service network and the theoretical proof of its scale-free characteristics. Enterp Inf Syst 6(4):373–404. doi:10.1080/17517575.2011.621981
25. Tao F, Zhang L, Lu K, Zhao D (2012) Research on manufacturing grid resource service optimal-selection and composition framework. Enterp Inf Syst 6(2):237–264. doi:10.1080/17517575.2010.540677
26. Tao F, Zhao D, Yefa H, Zhou Z (2010) Correlation-aware resource service composition and optimal-selection in manufacturing grid. Eur J Oper Res 201(1):129–143. doi:10.1016/j.ejor.2009.02.025
27. Tao F, Zhao D, Zhang L (2010) Resource service optimal-selection based on intuitionistic fuzzy set and non-functionality QoS in manufacturing grid system. Knowl Inf Syst 25(1):185–208. doi:10.1007/s10115-009-0263-6
28. Wang XV, Wang L (2014) From cloud manufacturing to cloud remanufacturing: a cloud-based approach for WEEE recovery. Manuf Lett 2(4):91–95. doi:10.1016/j.mfglet.2014.06.002
29. Wu Z, Gao Z, Cao Y, Ye X, Yang J (2015) Tolerance design and adjustment of complex customized product based on cloud manufacturing. Procedia CIRP 27:169–175. doi:10.1016/j.procir.2015.04.061
30. Tao F, Zhang L, Liu Y, Cheng Y, Wang L, Xu X (2015) Manufacturing service management in cloud manufacturing: overview and future research directions. J Manuf Sci Eng. doi:10.1115/1.4030510
31. He W, Xu L (2014) A state-of-the-art survey of cloud manufacturing. Int J Comput Integr Manuf 28(3):239–250. doi:10.1080/0951192X.2013.874595
32. Herroelen W, Leus R (2005) Project scheduling under uncertainty: survey and research potentials. Eur J Oper Res 165(2):289–306. doi:10.1016/j.ejor.2004.04.002
33. Maravas A, Pantouvakis J-P (2013) Guidelines for modelling time and cost uncertainty in project and programme management. Procedia Soc Behav Sci 74:203–211. doi:10.1016/j.sbspro.2013.03.045
34. Rand GK (2000) Critical chain: the theory of constraints applied to project management. Int J Proj Manag 18(3):173–177. doi:10.1016/S0263-7863(99)00019-8
35. Dixit V, Srivastava RK, Chaudhuri A (2014) Procurement scheduling for complex projects with fuzzy activity durations and lead times. Comput Ind Eng 76:401–414. doi:10.1016/j.cie.2013.12.009
36. Masmoudi M, Haït A (2013) Project scheduling under uncertainty using fuzzy modelling and solving techniques. Eng Appl Artif Intell 26(1):135–149. doi:10.1016/j.engappai.2012.07.012
37. Shukla S, Son Y, Tiwari MK (2008) Fuzzy-based adaptive sample-sort simulated annealing for resource-constrained project scheduling. Int J Adv Manuf Technol 36(9–10):982–995. doi:10.1007/s00170-006-0907-6
38. Ke H, Liu B (2010) Fuzzy project scheduling problem and its hybrid intelligent algorithm. Appl Math Model 34(2):301–308. doi:10.1016/j.apm.2009.04.011

39. Wang J (2005) Constraint-based schedule repair for product development projects with time-limited constraints. Int J Prod Econ 95(3):399–414. doi:10.1016/j.ijpe.2004.02.005
40. Zhu G, Bard JF, Yu G (2005) Disruption management for resource-constrained project scheduling. J Oper Res Soc 56(4):365–381
41. Van de Vonder S, Demeulemeester E, Herroelen W, Leus R (2005) The use of buffers in project management: the trade-off between stability and makespan. Int J Prod Econ 97(2):227–240. doi:10.1016/j.ijpe.2004.08.004
42. Lambrechts O, Demeulemeester E, Herroelen W (2008) Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities. J Sched 11(2):121–136. doi:10.1007/s10951-007-0021-0
43. Tao F, Hu Y, Zhao D, Zhou Z (2008) Study of failure detection and recovery in manufacturing grid resource service scheduling. Int J Prod Res 48(1):69–94. doi:10.1080/00207540802275871
44. Zhou J-T, Wang M-W, Yang H-C, Li M (2012) Capability driven project monitoring and management mechanism for cloud manufacturing. Comput Integr Manuf Syst 18(7):1518–1526
45. Li X, Zhou X, Ruan X (2002) Conflict management in closely coupled collaborative design system. Int J Comput Integr Manuf 15(4):345–352. doi:10.1080/09511920210121259
46. Klein M (1991) Supporting conflict resolution in cooperative design systems. IEEE Trans Syst Man Cybern 21(6):1379–1390. doi:10.1109/21.135683
47. Matta N, Corby O (2000) Conflict management in concurrent engineering: modelling guides. In: Dieng R (ed) Müller H. Computational Conflicts, Springer Berlin Heidelberg, pp 125–143. doi:10.1007/978-3-642-56980-7_7
48. Chen Y-M, Wang S-C (2007) An agent-based evolutionary strategic negotiation for project dynamic scheduling. Int J Adv Manuf Technol 35(3–4):333–348. doi:10.1007/s00170-006-0830-x
49. Ross S, Fang L, Hipel KW (2002) A case-based reasoning system for conflict resolution: design and implementation. Eng Appl Artif Intell 15(3–4):369–383. doi:10.1016/S0952-1976(02)00065-9
50. Guo S, Du B, Peng Z, Huang X, Li Y Manufacturing resource combinatorial optimization for large complex equipment in group manufacturing: a cluster-based genetic algorithm. Mechatronics (0). doi:10.1016/j.mechatronics.2015.03.005
51. Fei T, Ying Z, Da Li X, Lin L, Lin Z (2014) Internet of things and BOM-based life cycle assessment of energy-saving and emission-reduction of products. IEEE Trans Eng Manag 10(2):1252–1261. doi:10.1109/TII.2014.2306771
52. Magnaye R, Sauser B, Patanakul P, Nowicki D, Randall W (2014) Earned readiness management for scheduling, monitoring and evaluating the development of complex product systems. Int J Proj Manag 32(7):1246–1259. doi:10.1016/j.ijproman.2014.01.009
53. Dedehayir O, Nokelainen T, Mäkinen SJ (2014) Disruptive innovations in complex product systems industries: a case study. J Eng Technol Manag 33:174–192. doi:10.1016/j.jengtecman.2014.06.002
54. Ling W, Huanyu Z, Xiaolong Z (2014) Survey on resource-constrained project scheduling under uncertainty. Control Decis 29(4):577–584. doi:10.13195/j.kzyjc.2013.1309
55. Hu X-B, Wang M, Ye Q, Han Z, Leeson MS (2014) Multi-objective new product development by complete Pareto front and ripple-spreading algorithm. Neurocomputing 142:4–15. doi:10.1016/j.neucom.2014.02.058
56. Tao F, Feng Y, Zhang L, Liao TW (2014) CLPS-GA: a case library and Pareto solution-based hybrid genetic algorithm for energy-aware cloud service scheduling. Appl Soft Comput 19:264–279. doi:10.1016/j.asoc.2014.01.036
57. Tao F, Qiao K, Zhang L, Li Z, Nee AYC (2011) GA-BHTR: an improved genetic algorithm for partner selection in virtual manufacturing. Int J Prod Res 50(8):2079–2100. doi:10.1080/00207543.2011.561883
58. Tao F, Zhang L, Zhang ZH, Nee AYC (2010) A quantum multi-agent evolutionary algorithm for selection of partners in a virtual enterprise. CIRP Ann Manuf Technol 59(1):485–488. doi:10.1016/j.cirp.2010.03.120
59. Merkle D, Middendorf M, Schmeck H (2002) Ant colony optimization for resource-constrained project scheduling. IEEE Trans Evol Comput 6(4):333–346. doi:10.1109/TEVC.2002.802450
60. Peteghem VV, Vanhoucke M (2010) A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem. Eur J Oper Res 201(2):409–418. doi:10.1016/j.ejor.2009.03.034
61. Fei T, Yuanjun L, Yilong L, Ying F, Qining W, Lin Z, Xu L (2014) Concept, principle and application of dynamic configuration for intelligent algorithms. IEEE Syst J 8(1):28–42. doi:10.1109/JSYST.2013.2275619
62. Tao F, Li C, Liao TW, Laili Y (2015) BGM-BLA: a new algorithm for dynamic migration of virtual machines in cloud computing. IEEE Trans Serv Comput 99:1–1. doi:10.1109/TSC.2015.2416928