

Mi Liu, Shuping Yi\*, Peihan Wen and Haicao Song

# Disruption Management for Predictable New Job Arrivals in Cloud Manufacturing

DOI 10.1515/jisys-2016-0016

Received February 18, 2016; previously published online September 24, 2016.

**Abstract:** Manufacturing resources are shared and centrally managed on the cloud platform in cloud manufacturing, which is a new model of modern manufacturing. The production data are collected, which can be used to predict the manufacturing events. Based on those, disruption problems of scheduling should be researched from a new point of view. In this paper, new job arrivals were considered as the disruption event. The time of the occurrence of disruption was predictable in contrast to uncertainty. Alternative subcontractors chosen from the cloud platform were available for outsourcing with different processing prices and transporting distances. The objective of the original scheduling, the deviation between the new schedule and the old one, and the outsourcing cost were all considered. To express the problem, mathematical models and a three-field notation model were constructed. To solve the problem, a hybrid quantum-inspired chaotic group leader optimization algorithm was proposed, in which a hybrid encoding way was applied. To verify the algorithm, experiments were carried out. The results showed that the proposed algorithm performs well.

**Keywords:** Cloud manufacturing, disruption management, quantum-inspired evolutionary algorithm, group leader optimization algorithm, outsourcing.

**Classification according to MSC 2010.** 90B50.

## 1 Introduction

Production scheduling is one of the most important manufacturing management processes. The scheduling problem is considered to be dynamic in nature. For example, new orders may arrive every day. The new orders need to be integrated with the existing production schedule immediately with minimal disturbance on the performance and maximum stability of the existing schedule [25]. To solve such problems, disruption management is applied, which has a huge potential and offers substantial gains in efficiency for the users involved [3].

Conventionally, machine breakdowns [5, 7, 18] and new job arrivals [2, 24, 27] were the most common disruptions. The disruptions may disrupt the original schedule and cause a change to the original scheduling plan. Disruption management in scheduling has been studied deeply. Usually, make span [17] was taken as the criterion of scheduling efficiency and the cost of disruption [24] is measured in terms of the maximum difference of the original jobs before and after disruption. Meeting the due dates is another one of the most important objectives in scheduling [20].

Qi et al. [19] proposed a general framework of scheduling under either a random or an anticipated disruption with detailed analysis of shortest processing time (SPT) schedules. They analyzed both single-machine

---

\*Corresponding author: Shuping Yi, College of Mechanical Engineering, Chongqing University, Shazheng Street, Shapingba District, Chongqing 400044, China, e-mail: ysp\_cqu@sina.com

Mi Liu and Peihan Wen: College of Mechanical Engineering, Chongqing University, Chongqing 400044, China.

<http://orcid.org/0000-0001-5587-3966> (M. Liu); <http://orcid.org/0000-0002-7228-0000> (P. Wen)

Haicao Song: College of Mechanical Engineering, Chongqing University, Chongqing 400044, China; and College of Mechanical and Electrical Engineering, Shihezi University, Xinjiang 832003, China

and parallel two-machine problems using several deviation cost measures and rescheduling policies. The significant result they found was that the optimal schedule obtained after the disruption was still in SPT order in some form.

Similar to the SPT rule, the weighted discounted SPT (WDSPT) rule is employed to optimize disruptions to scheduling problems. With the measurement of the original objective and deviation, Liu et al. [18] used the WDSPT rule to study single-machine disruption problems, and found that when the disruption occurs, the machine disruption problem for the initial WDSPT schedule was still existing and therefore similar procedures could be implemented.

According to the dynamic feature of random job arrivals and machine breakdowns, Zhang et al. [27] proposed a hybrid genetic algorithm and a tabu search algorithm with a new initialization method to solve the problem. They also designed a simulator to generate the unexpected disruptions rather than the mathematical model, which is not easy to use to express the disruptions. The performance measures investigated were mean flow time, maximum flow time, mean tardiness, maximum tardiness, and the number of tardy jobs [27].

Excepting from machine breakdowns and new job arrivals, Katragjini et al. [13] employed another disruption named job ready time variations, which interrupt the original schedules simultaneously. Wang et al. [22] studied a rescheduling problem in response to the arrival of new jobs in a single-machine layout, where preventive maintenance was determined. Controllable processing time was an important factor affecting the rescheduling problems' flexibility under a single-machine breakdown [1]. With controllable processing times, anticipative decision making in preparing initial schedules can avoid excessive rescheduling costs, which may result from reactive processing time adjustments [9].

Cloud manufacturing (CMfg) is a new model of modern manufacturing. CMfg means a computing and service-oriented manufacturing model under the support of cloud computing, Internet of things (IoT), virtualization and service-oriented technologies, and advanced computing technologies. By providing safe and reliable, high-quality, cheap, and on-demand used manufacturing services for the whole lifecycle of manufacturing, CMfg aims to realize the full sharing and circulation, high utilization, and on-demand use of various manufacturing resources and capabilities [21]. Through the cloud platform, idle manufacturing resources of different manufacturers are shared. Thus, the utilization efficiency of total resources in society can be improved, and the overall cost of manufacturing can be reduced. In CMfg, the product sold to the world is supervisory controlled by the control center of the manufacturing. The equipment usage activity can be monitored in real time, and the breakdowns can be predicted [16] through technologies of IoT, big data analysis, cloud computing, etc.

According to the interviews made in a large equipment manufacturing plant with small batch production, we learned that components and parts of sold equipment may be destroyed because of the product life cycle and some random factors; thus, new orders of the replacing components and parts are accepted frequently. Those new orders are relatively large disturbances for manufacturing, which increase the difficulty of scheduling management and production costs. Thus, research on disruption management for predictable new job arrivals in CMfg is necessary.

As the disruption can be predicted and the available manufacturing resources can be obtained on the cloud platform in real time, the manufacturer can take measures to deal with the disruption in advance. That is, specific information about the new jobs and the time point of disruption occurrence is known in advance, and the manufacturer could decide which subcontractor is chosen for outsourcing and which jobs should be outsourced. Therefore, part of the risk of production cost and tardiness will be transferred to the subcontractor. On the basis of the review mentioned above, we can know that the objectives of the disruption were always set from two aspects. One was the original objective function. The other was the cost of deviation from the original schedule. Based on the two criteria, we consider the objective of the disruption management cost in addition. In this paper, the disruption of predictable new job arrivals in CMfg environment is studied. Specific information about the new jobs and the time point of disruption occurrence is known in advance. To reduce the cost resulting from disruption and the deviation from original schedule for the largest possible reduction, two subcontractors from the cloud platform

are available for outsourcing. Mathematical models and a three-field notation model are constructed to describe the problem. To solve those models, a quantum-inspired chaotic group leader optimization algorithm (QCGLOA) is proposed.

The rest of this paper is organized as follows. The problem is described in the next section. Section 3 contains the management strategies and mathematical models. In Section 4, a hybrid algorithm to solve the model is proposed. In Section 5, the experimental results and discussions are shown. Section 6 gives the conclusions.

## 2 Problem Description and Notations

The problem is stated as follows. A manufacturer needs to process a set of jobs with the objective of minimizing the job completion time. An original schedule  $\pi$  is produced in the SPT first rule [19]. Before the original and optimal plan is carried out, another set of new jobs is predicted to arrive at  $t_0$  during the planning horizon. The new jobs have no preemption. At the arrival time, there are  $n$  jobs waiting to be processed according to the original schedule. We consider the  $n$  jobs as old job set  $JO = \{J_1, J_2, \dots, J_n\}$ , and the  $\tilde{n}$  new arrival jobs as new job set  $JN = \{J_{n+1}, J_{n+2}, \dots, J_{n+\tilde{n}}\}$ .  $JO$  and  $JN$  are also arranged in SPT rule. There is no idle time between two jobs. The number of all the jobs is  $N = n + \tilde{n}$ , and the set of the jobs is described as  $J = JO \cup JN$ .

Usually, the new jobs are processed immediately after the original schedule completed. This can ensure that the old jobs could be delivered on time. However, the new jobs have a likewise due date. If the new ones are processed after all the old ones completed, tardiness penalties would be incurred and the production cost would be increased. In CMfg, utilizing the cloud platform, the manufacturer can outsource the tasks to the subcontractors. In doing so, the risk of cost increase is transferred.

Now, it comes to the problem of which jobs are more suitable for outsourcing, and which subcontractor is chosen to accept the tasks. If the jobs are from  $JO$ , they should be shipped before the original schedule is executed. If the jobs are from  $JN$ , the jobs should be shipped to the subcontractor directly, rather than to the manufacturer. Then, the new schedule  $\pi^*$  is generated. We assume that the new jobs arrive at the subcontractor's shop floor at  $t_0$ , and the number of alternative subcontractors is 2.

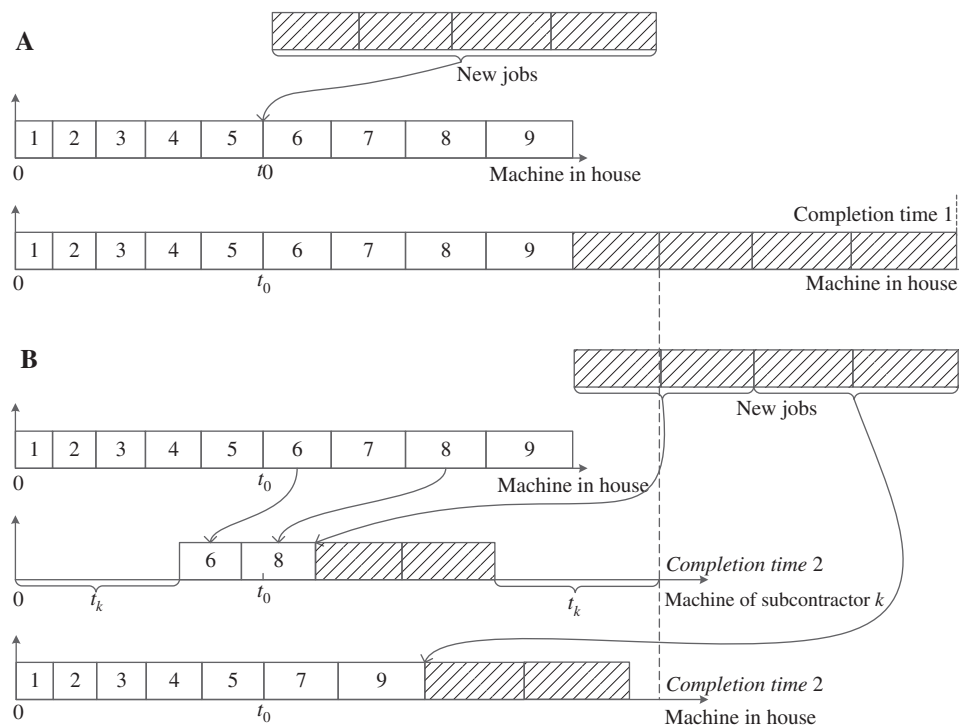
For the problem, we use the following notations:

- $e_i$ : the processing time of  $J_i$  processed in house,  $i = 1, 2, \dots, n + \tilde{n}$ ;
- $e_{ik}$ : the processing time of  $J_i$  processed by subcontractor  $k$ ,  $i = 1, 2, \dots, n + \tilde{n}$ ,  $k = 1, 2$ ;
- $C_i$ : the original completion time of  $J_i$  processed in schedule  $\pi$ ,  $i = 1, 2, \dots, n$ ;
- $Cr_i$ : the completion time of  $J_i$  in the new schedule  $\pi^*$ ,  $i = 1, 2, \dots, n + \tilde{n}$ ;
- $dd_i$ : the delivery date of  $J_i$  processed in house,  $i = 1, 2, \dots, n + \tilde{n}$ ;
- $v_i$ : the value of  $J_i$ ,  $i = 1, 2, \dots, n + \tilde{n}$ ;
- $p_i$ : the unit penalty of  $J_i$  for lateness,  $i = 1, 2, \dots, n + \tilde{n}$ ;
- $D_k$ : the distance from manufacturer to subcontractor  $k$ ,  $k = 1, 2$ ;
- $CD_k$ : the cost of the unit distance of subcontractor  $k$ ,  $k = 1, 2$ ;
- $t_k$ : the transporting time from manufacturer to subcontractor  $k$ ,  $k = 1, 2$ ;

Let deviation  $T_i = \max\{Cr_i - C_i, 0\}$ , transportation cost  $TC_k = D_k * CD_k$ . The two indices can be used to assess the deviation and disruption cost.

## 3 Strategies and Mathematical Models

To describe the problem more clearly, the strategy of disruption management mentioned above is shown in Figure 1. Figure 1A displays that the new jobs without preemption are predicted to arrive at  $t_0$ . They are pro-



**Figure 1:** (A) Usual Practice of Disruption of New Job Arrivals. (B) Outsourcing for New Job Arrivals in CMfg.

cessed after all the old jobs completed, as the usual practice. The total completion time is *completion time 1*. In Figure 1B, subcontractor  $k$  is chosen to share the risk of lateness and tardiness. Specifically, jobs 6 and 8 are chosen for outsourcing at  $t = 0$ . The shipping time is  $t_k$ . Some of the new jobs are selected to process in house, and the left ones, shipped to the subcontractor directly, are processed through outsourcing. They are predicted to arrive at the manufacturer in house and at the subcontractor's job shop at  $t_0$ , respectively. The new jobs are processed after the old ones completed. Then, all the jobs for outsourcing are shipped back to the manufacturer in house. The total completion time is *completion time 2*. *completion time 2* would be sharply decreased compared with *completion time 1*.

Typically, decision models used in disruption management employ two criteria. The first one is the original objective function and the second is a measure of the deviation from the original schedule [19]. Different from most studies, the cost brought by the disruption is considered as the third criterion in this paper, which was inspired by Ref. [14]. The disruption cost is evaluated by the transportation cost, which involves the production cost of the subcontractors' machines.

Considering the objective of minimum tardiness time, the first function is defined as

$$f_1 = \min \sum_{i=1}^{n+\bar{n}} Cr_i. \quad (1)$$

Considering the objective of minimum deviation, the second function is defined as

$$f_2 = \min \sum_{i=1}^n T_i. \quad (2)$$

Considering the objective of minimum disruption cost, the third function is defined as

$$f_3 = \min \sum_{k=1,2} TC_k. \quad (3)$$

A weighted function is employed to aggregate the multiple objectives into a single one; thus, the three-objective problem can be converted to a single-objective problem [15]. Let  $\lambda_1, \lambda_2, \lambda_3$  be the weights of each function,  $0 < \lambda_1, \lambda_2, \lambda_3 < 1, \lambda_1 + \lambda_2 + \lambda_3 = 1$ . The single-objective problem is expressed as

$$f = \lambda_1 * f_1 + \lambda_2 * f_2 + \lambda_3 * f_3. \quad (4)$$

The three objectives are subjected to the tardiness penalty, which is represented as

$$(Cr_i - dd_i) * p_i < v_i. \quad (5)$$

Furthermore, the completion time of the old jobs for outsourcing is the time all the jobs shipped back to manufacturer, so

$$Cr_i = Cr_n + t_k, \quad (6)$$

$$Cr_n = Cr_{n-1} + e_{nk}, \quad (7)$$

where  $i = 1, 2, \dots, n, Cr_1 = t_k + e_{1k}$ . In like manner, the completion time of the new jobs for outsourcing is

$$Cr_j = Cr_{n+\tilde{n}} + t_k, \quad (8)$$

$$Cr_{n+\tilde{n}} = Cr_{n+\tilde{n}-1} + e_{ik}, \quad (9)$$

where  $j = n+1, \dots, n+\tilde{n}, Cr_{n+1} = Cr_n + e_{(n+1)k}$ .

For the constrained optimization, we introduce the penalty function to transfer the constrained problem to an unconstrained one. The fitness function is

$$F = f + \mu * (\max(0, \zeta))^2, \quad (10)$$

where  $\mu$  is a large positive number,  $\zeta = (Cr_i - dd_i) * p_i - v_i$ .

The well-known three-field notation  $\alpha|\beta|\gamma$  scheme [8] for scheduling problems is adopted in this paper, where  $\alpha$  indicates the machine or scheduling environment,  $\beta$  describes the job characteristics or restrictive requirements, and  $\gamma$  defines the objectives to be minimized. We consider a two-machine problem, thus  $\alpha = 2$ . The disruption management strategy employed in this paper is a predictive management, and the disruption of non-preemptive job arrivals is predicted to happen at time  $t$ ; thus,  $\beta$  is represented as  $t, npmtn, pred - mgt$ . Therefore, the proposed problem is expressed as

$$2|t, npmtn, pred - mgt| \sum Cr_i, \sum T_i, \sum TC_k \quad (11)$$

where  $\sum Cr_i, \sum T_i, \sum TC_k$  means it is a tri-criteria scheduling problem.

**Theorem 1.** *The proposed problem  $2|t, npmtn, pred - mgt| \sum Cr_i, \sum T_i, \sum TC_k$  is NP-hard.*

*Proof.* This can be proved by reducing from the three-partition problem, which is known to be NP-complete; please refer to Ref. [14].  $\square$

**Theorem 2.** *For problem  $2|t, npmtn, pred - mgt| \sum Cr_i, \sum T_i, \sum TC_k$ , there exists an optimal schedule  $\pi^*$  in which the jobs of JO are sequenced in SPT order as in  $\pi$ , the jobs of JN are sequenced in SPT order, and there is no idle time between jobs [10].*

*Proof.* First, we analyze the jobs of JO. Suppose  $\sigma^*$  is an optimal schedule in which the jobs of JO are not sequenced in SPT order as in  $\pi$ . Let  $i$  be the job with the smallest index that appears later relative to the other jobs of JO in  $\sigma^*$  than in  $\pi$ . Let  $j > i$  be the immediate predecessor job of job  $i$  of JO in  $\sigma^*$ . Because  $\pi$  is an SPT schedule,  $e_i < e_j$ . Now, interchange jobs  $i$  and  $j$ , and obtain a new schedule  $\sigma'$ . In  $\sigma'$ ,  $Cr_i(\sigma') = Cr_j(\sigma') - p_j + p_i \leq Cr_j(\sigma')$ .  $Cr_j(\sigma') = Cr_i(\sigma')$ . All jobs between  $j$  and  $i$  are completed  $p_j - p_i \geq 0$  units earlier in  $\sigma'$  than in  $\sigma^*$ . A straightforward computation shows that the total completion time does not increase as a result of the interchange. If  $Cr_j(\sigma') \geq Cr_j(\pi)$  (as is the case for job  $i$ ), then  $Dev_j(\pi, \sigma') < Dev_j(\pi, \sigma^*)$ ; otherwise,  $Dev_j(\pi, \sigma') < Dev_j(\pi, \sigma^*)$ , where  $Dev_j(v, n)$  is the time disruption of job  $j, j \in JO$ . Thus,  $\sigma'$  is feasible and optimal. A finite number of repetitions

of this argument show that there exists an optimal schedule  $\pi^*$  in which the jobs of  $JO$  are sequenced in SPT order as in  $\pi$ . Similarly, it can be proved that jobs in  $JN$  are sequenced in SPT order. The same SPT ordering of the jobs in  $JO$  in  $\pi$  and  $\pi^*$  shows that there is no idle time in  $\pi^*$ . Otherwise, removing this idle time maintains feasibility and decreases the total completion time.  $\square$

As stated above, Eq. (11) is essentially to solve the problem of which jobs are made in house, which ones are made outsourced, and which subcontractor should be chosen. The solution can be represented as two binary strings:  $x = (x(1), x(2), \dots, x(i), \dots, x(n + \tilde{n}))$  and  $y = (y(1), y(2))$ , where  $x$  denotes the jobs and  $y$  denotes the subcontractors. The representing way of solutions will be applied in the proposed algorithm.

## 4 Proposed Algorithm

Equation (11) has been proved to be NP-hard. Consequently, a hybrid heuristic algorithm named QCGLOA is proposed to solve it. This section first introduces the quantum-inspired evolutionary algorithm (QIEA) and group leader optimization algorithm (GLOA), respectively. Then, the customized hybrid algorithm is described in detail.

### 4.1 Quantum-Inspired Evolutionary Algorithm

QIEA was adapted from quantum computing in physics, first proposed by Han and Kim [11]. QIEA is characterized by the representation of the individual, the evaluation function, and the population dynamics, like other evolutionary algorithms. However, instead of binary, numeric, or symbolic representation, Q-bit representation is used. What is more, a Q-gate is used as the evolutionary operator, and an observation process is employed to connect the Q-bit representation with the optimization variables [26].

A Q-bit is defined with a pair of complex numbers  $(\alpha, \beta)$  as  $[\alpha, \beta]^T$ , where  $|\alpha|^2 + |\beta|^2 = 1$ .  $|\alpha|^2$  gives the probability that the Q-bit will be found in the “0” state, and  $|\beta|^2$  gives the probability that the Q-bit will be found in the “1” state.

A Q-bit individual is defined as a string of  $m$  Q-bits. It can be defined as

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{bmatrix}, \quad (12)$$

where  $|\alpha_j|^2 + |\beta_j|^2 = 1, j = 1, 2, \dots, m$ .

Q-bit representation has a better characteristic of population diversity than other representations. That is because Q-bit is the smallest unit of information as a probabilistic representation. It may be in the “1” state, in the “0” state, or in any superposition of the two. Thus, a Q-bit individual can represent a linear superposition of states in search space probabilistically. Each Q-bit can be rendered into one binary bit by employing the probabilistic observation process. Thus, the number of the linear superposition of states is  $2^m$ .

For example, when  $m = 3$  in Eq. (12), the three-Q-bit string may be

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{\sqrt{15}}{4} & \frac{1}{2} \\ 2\sqrt{\frac{2}{3}} & -\frac{1}{4} & \sqrt{\frac{3}{2}} \end{bmatrix}. \quad (13)$$

For each Q-bit, a very small value  $\eta_j, \eta_j \in [0, 1]$  is generated randomly. If  $\eta_j < |\alpha_j|^2$ , the observation value of the  $j$ th Q-bit is 0. Otherwise, it is 1.  $|\beta_j|^2 = 1 - |\alpha_j|^2$ . The observed superposition of Eq. (13) will be  $|010\rangle, |011\rangle, |000\rangle, |001\rangle, |110\rangle, |111\rangle, |100\rangle, |101\rangle$ , and the number of superposition of states is  $2^3 = 8$ .

A Q-gate is used to modify the state of a Q-bit, playing the role of an evolutionary operator. There are various gates, such as the rotation gate, NOT gate, AND gate, OR gate, NAND gate, and Hadamard gate [12]. Rotation Q-gate is employed in this paper.



## 4.2 Group Leader Optimization Algorithm

GLOA was first proposed by Anmer and Sabre in 2011 [4]. It is a new global optimization evolutionary algorithm that was inspired by the leaders' social characteristics. Leaders in social groups command members to complete tasks. The members' characteristics and performance are impacted by the leaders. When a member's leadership is trained to be better than the old leader, the new one is substituted for the old one. In standard GLOA, there are six steps constituting a continuous iteration and cycle process [23].

Step 1. Generate  $g$  groups and  $m$  members for each group randomly.

Step 2. Calculate fitness values for all members in all groups.

Step 3. Appoint the leader for each group. A member with the best fitness value in the group is the leader.

Step 4. Mutation and recombination. This step is aimed to create new excellent members by using the old members, group leader, and a random element. The expression is as follows:

$$new = r_1 * x + r_2 * leader + r_3 * random, \quad (14)$$

where  $r_1$  is the genetic probability of old member,  $r_2$  is the impact factor of group leader,  $r_3$  is the random factor, and  $x$  is the member to be mutated, called old member.  $r_1 + r_2 + r_3 \leq 1$ . If the new member's fitness value is better than the old one, then the new one replaces the old one. Otherwise, retain the old member.

Step 5. One-way crossover. This operator is performed from the first group to the last one. Choose a member in the group randomly, and then transfer some of its parameters from another random member in another random group. If the generated new member has a better fitness value, then substitute it for the old one. Otherwise, keep the old member.

Step 6. Repeat step 3 to step 5 according to the number of given iteration times.

GLOA is able to search the solution space surrounding the leaders, which are possibly local or global minima. The members of the groups are not subjected to a local minimization. Therefore, GLOA allows the population to converge upon global minima in a very fast way.

## 4.3 Quantum-Inspired Chaotic Group Leader Optimization Algorithm

Q-bit representation in QIEA has the good characteristic of population diversity, while GLOA has good global search ability and chaotic mapping has the characteristic of randomness. Therefore, we tried to apply the advantages of the three algorithms to solve the proposed problem. The integrated framework is illustrated in Figure 2.

The procedure of QCGLOA can be summarized as follows.

1. Generate the initial population randomly based on Q-bit representation.
2. Convert each individual from Q-bit representation to a binary one by observing the superposition of states.
3. Evaluate each individual in the multi-objective sense.
4. Assign the leader of each group with minimum fitness.
5. Evolve the population toward to group leader for each group.
6. Output the best solution if the termination condition is satisfied.

### 4.3.1 Population Initialization and Representation

Suppose the generation is  $it$ . Let  $it=0$  and randomly generate an initial population  $GRP_g(it) = \{Mem_{g1}^{it}, \dots, Mem_{gm}^{it}, \dots, Mem_{gM}^{it}\}$ ,  $g=1, 2, \dots, G$ ,  $m=1, 2, \dots, M$ , where  $GRP_g(it)$  denotes the  $g$ th group in the  $it$  th generation and  $Mem_{gm}^{it}$  denotes the  $m$ th member in the  $g$ th group with the Q-bit representation

$$Mem_{gm}^{it} = \begin{bmatrix} \alpha_{g1}^{it} & \alpha_{g2}^{it} & \dots & \alpha_{gn}^{it} & \alpha_{gn+1}^{it} & \dots & \alpha_{gn+\bar{n}}^{it} & \alpha_{gn+\bar{n}+1}^{it} \\ \beta_{g1}^{it} & \beta_{g2}^{it} & \dots & \beta_{gn}^{it} & \beta_{gn+1}^{it} & \dots & \beta_{gn+\bar{n}}^{it} & \beta_{gn+\bar{n}+1}^{it} \end{bmatrix}.$$

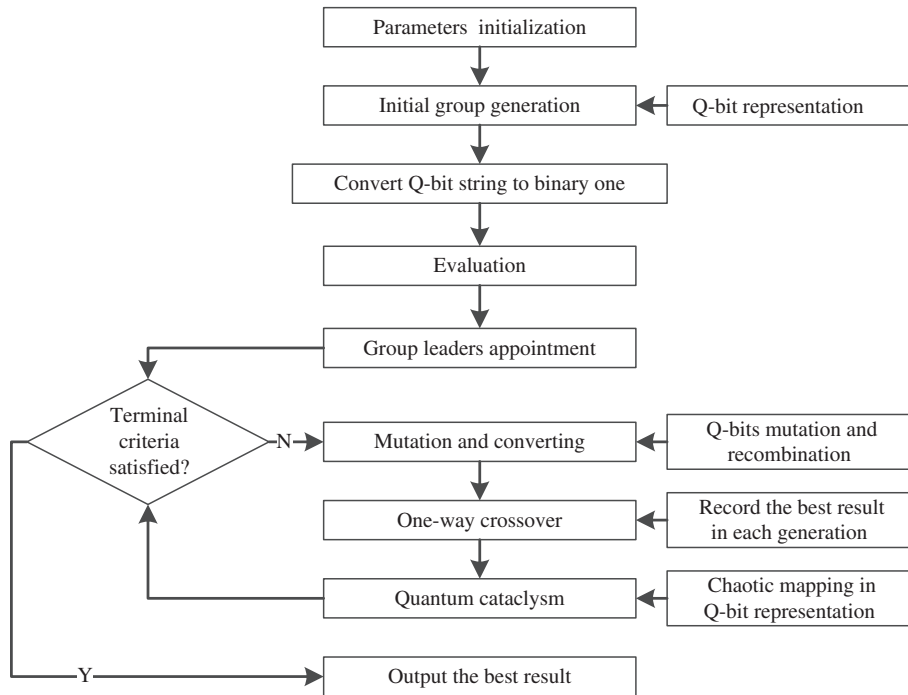


Figure 2: Integrated Framework of QCGLOA.

As shown, each member consists of  $n + \tilde{n} + 1$  bits. The bits  $1, \dots, n$  represent the old jobs, while bits  $n + 1, \dots, n + \tilde{n}$  are the arriving new jobs and the last bit in a member stands for the subcontractor. Each bit in the string denotes that the job is made in house or outsourced, and the outsourcing subcontractor is either  $k$  or the other one. To satisfy Eq. (11)'s objectives, the Q-bit representation will be converted to binary form.

#### 4.3.2 Converting

Because the objective-based jobs are represented in a binary system, the Q-bit cannot be used directly. So a converting mechanism needs to be introduced. The details of the mechanism are described as follows.

Observe the probability amplitudes  $\alpha$  and  $\beta$ . As described in Section 4.1, for each Q-bit, generate  $\eta_i$  randomly in  $[0, 1]$ . If  $\eta_i < |\alpha_i|^2$ , the  $i$ th position is converted to 0; otherwise, 1 is obtained. For the first part of a member, 1 means job  $i$  is made in house, and 0 means job  $i$  is made outsourced. For the second part of a member, 1 means the first subcontractor is selected; otherwise, the second one is chosen. For a 10-job problem, let  $n + \tilde{n} = 10$ .

The length of member is 11. A member is randomly generated as  $Mem_{gm}^{it} = \left[ \begin{array}{c|c|c|c} \frac{1}{\sqrt{4}} & \frac{1}{3} & \dots & \frac{1}{2} \\ 3 & 2\sqrt{2} & \dots & \sqrt{3} \\ \hline \frac{1}{\sqrt{4}} & \frac{1}{3} & \dots & \frac{1}{2} \end{array} \right]$ .

As one possible case, the string is converted to  $[1|1|0|0|1|0|1|1|0|0]$ . That is to say, jobs 1, 2, 3, 5, 7, 8, 9 are processed in house, and jobs 4, 6, 10 are outsourced. The last bit shows that subcontractor 2 is selected.

#### 4.3.3 Individual Evaluation

First, calculate every scheduling objective according to Eqs. (1)–(3) and (5)–(9). Subsequently, a vector of weights is randomly generated to evaluate individuals in a multi-objective sense according to Eqs. (4) and (10). An individual is also a solution. The smaller the fitness value is, the better the solution. The one with the smallest value in a group is regarded as the group leader, which will be employed to guide the GLOA mutation



in the current generation (see Section 4.3.4). As the weight vector is randomly generated, it is helpful to stress the search toward the direction determined by the weight vector.

#### 4.3.4 Mutation

In every generation and in each group, all the members except the leader are traveled by the sequential traversal method. A new member is decided by the traveled old member, leader, and a random variable. The pseudocode of mutation is as follows:

```

 $x \leftarrow$  the  $i$ th bit of a member
 $Mem_{gm} \leftarrow$  the  $m$ th member in group  $g$ 
For  $g = 1$  to  $G$  do
  For  $m = 1$  to  $M$  do
    For  $i = 1$  to  $n + \tilde{n}$  do
       $\alpha_{gm i new} = r_1 * \alpha_{gm i} + r_2 * \alpha_{leader i} + r_3 * random$ 
       $|\beta_{gm i new}|^2 = 1 - |\alpha_{gm i new}|^2$ 
      Random  $\eta_i$ 
      If  $\eta_i < |\alpha_{gm i new}|^2$ 
         $x \leftarrow 0$ 
      Else
         $x \leftarrow 1$ 
      End
    End
     $Mem_{gm new}$  is generated
    Calculate the fitness value of  $Mem_{gm new}$ 
    If fitness( $Mem_{gm new}$ ) is better than fitness( $Mem_{gm}$ )
       $Mem_{gm} = Mem_{gm new}$ 
    Else
       $Mem_{gm} = Mem_{gm}$ 
    End
  End
End

```

The following strategy is for group  $g$ , and other groups are executed in the same way.

Step 1: For job string, according to Eq. (9),  $\alpha_{m i new} = r_1 * \alpha_{m i} + r_2 * \alpha_{leader i} + r_3 * random$ , where  $i = 1, 2, \dots, n + \tilde{n}$ ;  $m = 1, 2, \dots, M$ ;  $m \neq leader$ .  $|\beta_{m i new}|^2 = 1 - |\alpha_{m i new}|^2$ . The operation is also applied to the subcontractor string. Then, a new member encoded in Q-bit way is generated.

Step 2: Calculate a new member's fitness value. If the value is better than the old one, then replace the old member by the new one. Otherwise, keep the old member in the group and proceed to the next one.

#### 4.3.5 One-Way Crossover

To escape local minimum, a one-way crossover operator is employed. It travels all the groups from the first group. When it goes into group  $g$ , member  $Mem_{gm}$  and its attributes are randomly selected at a certain probability. Subsequently, a member from another group is randomly selected, and its attributes in the same bits are transferred to  $Mem_{gm}$ . For each group, the times of transferring is  $\theta$ .  $1 \leq \theta \leq (\frac{\# parameter}{2}) + 1$ , where  $\# parameter$  is the number of bits of each member. Finally, a new member is generated. The subsequent steps are the same as those in mutation, e.g. evaluate and decide whether the old member will be replaced by the new one.



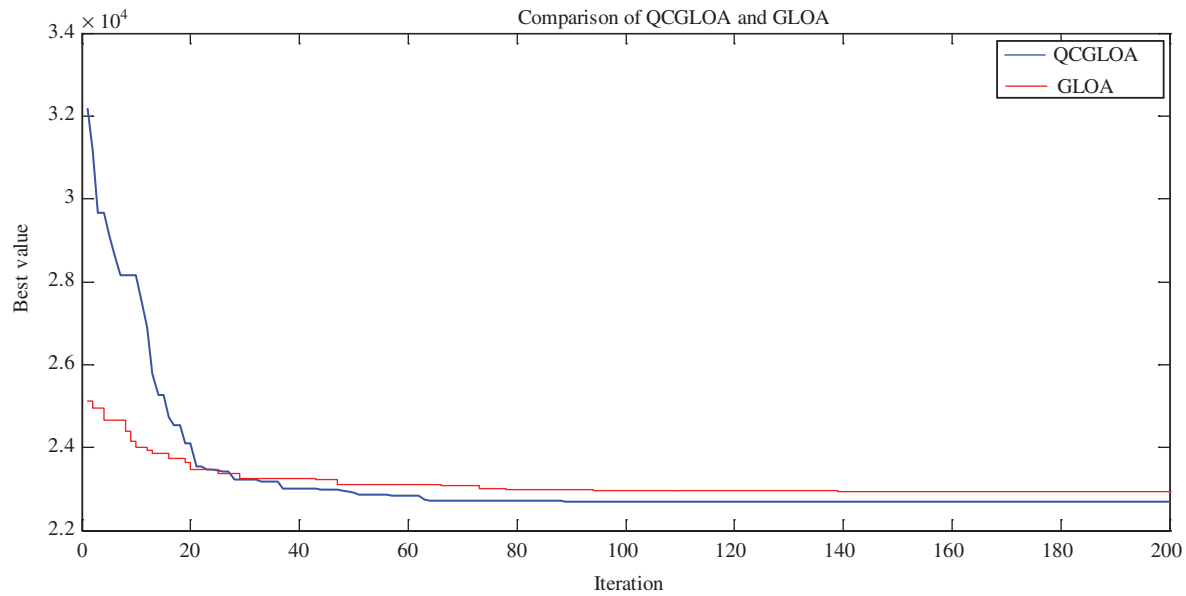


Figure 3: Simulation Result of QCGLOA Compared with GLOA.

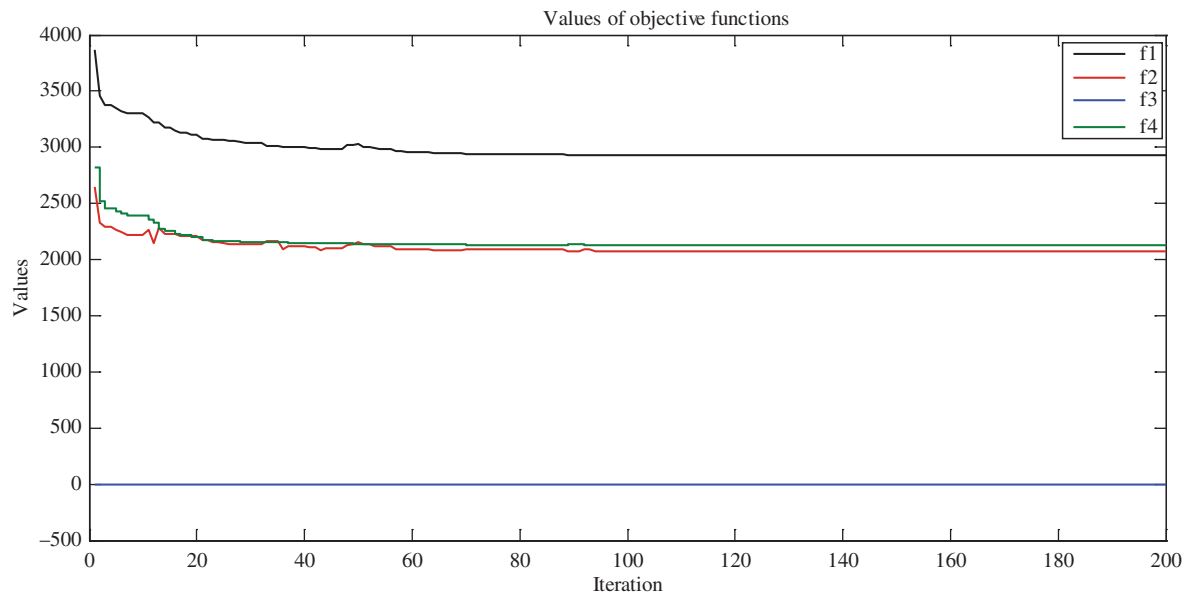


Figure 4: Objective Value of Each Independent Function.

## 6 Conclusions

This paper researched the disruption management of new job arrivals in CMfg, which is a new issue. The new jobs are the ones substituting for the components and parts of the sold products, the arrival time of which can be predicted. In CMfg, the information about manufacturing in real time of subcontractors can be obtained in cloud platform. The outsourcing of jobs and selection of subcontractors were discussed; moreover, mathematical models were constructed in this paper. The models involved the objective of original optimal scheduling, the deviation from the original scheduling plan to the revised one, as well as the

outsourcing cost measured by subcontractor selection. A QCGLOA was proposed to solve the models. A quantum-inspired evolutionary algorithm was employed to enhance the diversity of the population, while a group leader optimization algorithm was applied to improve the capacity of exploration and chaotic mapping was applied to regenerate the random groups. The experiments showed that the proposed algorithm has good performance.

The main contributions of this paper are as follows:

1. Constructing mathematical models to express the disruption problem of new job arrivals in CMfg with distributed manufacturing resources.
2. Expressing the jobs and the alternative subcontractors in one hybrid individual, which simplifies the way of coding.
3. Proposing a QCGLOA to solve the model. The hybrid algorithm adopted the advantages of three independent algorithms.

Although this paper has achieved the mentioned contributions, some work still has to be done. For example, the disruption management of machine breakdowns in CMfg should be studied, and the performance of the proposed algorithm in this paper should be improved in the future.

**Declaration of conflicting interests:** The authors declare that there is no potential conflict of interest.

**Funding:** This research was supported by National Natural Science Foundation of China (no. 71501020).

## Bibliography

- [1] M. S. Aktürk, A. Atamtürk and S. Gürel, Parallel machine match-up scheduling with manufacturing cost considerations, *J. Scheduling* **13** (2010), 95–110.
- [2] Y. Chiu and C. Shih, Rescheduling strategies for integrating rush orders with preventive maintenance in a two-machine flow shop, *Int. J. Prod. Res.* **50** (2012), 5783–5794.
- [3] J. Clausen, J. Larsen, A. Larsen and J. Hansen, Disruption management, *ORMS Today* **28** (2001), 40–43.
- [4] A. Daskina and S. Kais, Group leaders optimization algorithm, *Mol. Phys.* **109** (2011), 761–772.
- [5] Y. Dong and J. Jang, Production rescheduling for machine breakdown at a job shop, *Int. J. Prod. Res.* **50** (2012), 2681–2691.
- [6] A. H. Gandomi, G. J. Yun, X. S. Yang and S. Talatahari, Chaos-enhanced accelerated particle swarm optimization, *Commun. Nonlinear Sci.* **18** (2013), 327–340.
- [7] S. Goren, I. Sabuncuoglu and U. Koc, Optimization of schedule stability and efficiency under processing time variability and random machine breakdowns in a job shop environment, *Naval Res. Logist.* **59** (2012), 26–38.
- [8] R. L. Graham, E. L. Lawler, J. K. Lenstra and A. H. G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey, *Ann. Discrete Math.* **5** (1979), 287–326.
- [9] S. Gürel, E. Körpeoğlu and M. S. Aktürk, An anticipative scheduling approach with controllable processing times, *Comput. Oper. Res.* **37** (2010), 1002–1013.
- [10] N. G. Hall and C. N. Potts, Rescheduling for new orders, *Oper. Res.* **52** (2004), 440–453.
- [11] K. H. Han and J. H. Kim, Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, *IEEE T. Evol. Comput.* **6** (2002), 580–593.
- [12] T. Hey, Quantum computing: an introduction, *Comput. Control Eng. J.* **10** (1999), 105–112.
- [13] K. Katragjini, E. Vallada and R. Ruiz, Flow shop rescheduling under different types of disruption, *Int. J. Prod. Res.* **51** (2013), 780–797.
- [14] C. Y. Lee, J. Y. T. Leung and G. Yu, Two machine scheduling under disruptions with transportation considerations, *J. Scheduling* **9** (2006), 35–48.
- [15] B. B. Li and L. Wang, A hybrid quantum-inspired genetic algorithm for multiobjective flow shop scheduling, *IEEE Trans. Syst. Man Cybernet. B* **37** (2007), 576–591.
- [16] J. Li, K. Shuang, S. Su, Q. Huang, P. Xu, X. Cheng and J. Wang, Reducing operational costs through consolidation with resource prediction in the cloud, in: *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, IEEE Computer Society, Washington DC, USA, pp. 793–798, 2012, doi: 10.1109/CCGrid.2012.50.
- [17] L. Liu and H. Zhou, Single-machine rescheduling with deterioration and learning effects against the maximum sequence disruption, *Int. J. Syst. Sci.* **46** (2015), 2640–2658.

- [18] F. Liu, J. Wang and D. Yang, Solving single machine scheduling under disruption with discounted costs by quantum-inspired hybrid heuristics, *J. Manuf. Syst.* **32** (2013), 715–723.
- [19] X. Qi, J. F. Bard and G. Yu, Disruption management for machine scheduling: the case of SPT schedules, *Int. J. Prod. Econ.* **103** (2006), 166–184.
- [20] G. Steiner and R. Zhang, Revised delivery-time quotation in scheduling with tardiness penalties, *Oper. Res.* **59** (2011), 1504–1511.
- [21] F. Tao, L. Zhang, V. C. Venkatesh, Y. Luo and Y. Cheng, Cloud manufacturing: a computing and service-oriented manufacturing model, *Eng. Manuf.* **225** (2011), 1969–1976.
- [22] K. Wang, S. H. Choi, H. Qin and Y. Huang, A cluster-based scheduling model using SPT and SA for dynamic hybrid flow shop problems, *Int. J. Adv. Manuf. Tech.* **67** (2013), 2243–2258.
- [23] F. Xiang, Y. F. Hu, Y. R. Yu and H. C. Wu, QoS and energy consumption aware service composition and optimal-selection based on Pareto group leader algorithm in cloud manufacturing system, *Cent. Eur. J. Oper. Res.* **22** (2014), 663–685.
- [24] Z. Xingong and W. Yong, Rescheduling problems with agreeable job parameters to minimize the tardiness costs under deterioration and disruption, *Math Probl. Eng.* **2013** (2013), 1–7.
- [25] Z. Zakaria and S. Petrovic, Genetic algorithms for match-up rescheduling of the flexible manufacturing systems, *Comput. Ind. Eng.* **62** (2012), 670–686.
- [26] G. X. Zhang, Quantum-inspired evolutionary algorithms: a survey and empirical study, *J. Heuristics* **17** (2011), 303–351.
- [27] L. Zhang, L. Gao and X. Li, A hybrid intelligent algorithm and rescheduling technique for job shop scheduling problems with disruptions, *Int. J. Adv. Manuf. Tech.* **65** (2013), 1141–1156.