Journal of Computers Vol. 30 No. 5, 2019, pp. 60-74
doi:10.3966/199115992019103005005

Multi-objective Scheduling Optimization of Manufacturing Resources in Cloud Manufacturing Environment

Hong-Guo Zhang, Yan-Lei Shi, Chao Ma*, Shu-Li Zhang, Sheng-Hui Liu

Software & Microelectronics School, Harbin University of Science and Technology,
XueFu-Road-52, Harbin, Heilongjiang, China
{zhg07, shiyanlei92, machao8396, zhangshuli0523}@163.com, hrbust.lsh@126.com

Received 30 October 2017; Revised 18 April 2018; Accepted 2 June 2018

Abstract. For efficiently carrying out the "working procedure level" manufacturing resource scheduling from the perspective of cloud manufacturing service platform, considering not only the characteristics of distribution and diversity of manufacturing resource but also the constraints related to the service quality and the users' subjective evaluation, the multi-objective optimization model was presented based on the four objectives: minimizing the makespan and the total cost, maximizing the quality of cloud manufacturing resource service and the users' satisfaction. Then an improved non-dominated particle swarm optimization algorithm is designed for the model. In the algorithm, on the one hand the dynamic learning factor is introduced to guide the search scope and precision of the particles, so as to avoid falling into the local optimal solution, on the other hand the dynamic selection factor is introduced to improve the diversity of population, and so as to enable the algorithm converges to the optimal solution. Finally, the experiment results demonstrate the feasibility and effectiveness of the above model and algorithm.

Keywords: cloud manufacturing, manufacturing resource scheduling, multi-objective optimization, particle swarm optimization

1 Introduction

In the condition of unbalanced distribution of manufacturing resources, on the one hand small and medium-sized enterprises lack of critical processing equipment are not capable enough to complete large-scale and complex processing tasks, and on the other hand, large enterprises are faced with the problem of waste caused by idle manufacturing resources. All of these make the traditional manufacturing method and manufacturing mode urgently need to be upgraded and transformed. Li proposed an advanced manufacturing model "cloud manufacturing", and put forward the viewpoint of "manufacturing as a service" [1-2]. Cloud manufacturing is based on the development of networked manufacturing. The networked manufacturing integrates the scattered resource to build the virtual resource pool and focus on using them. This is one mode that is use of scattered resource in a focused and effective manner.

The cloud manufacturing not only inherits this mode, but also realizes another mode that is use of centralized resource in the manner of service call or resource rent. The manufacturing resources located in different enterprises are encapsulated as set of cloud services and published to the cloud manufacturing platform. And then the corresponding manufacturing solutions composing of a series of cloud services are provided dynamically for users according to their different requirements [3].

For enterprises that are not able to complete the processing tasks independently, they can accomplish the tasks through rational scheduling of manufacturing resource from multi-enterprises. Due to the large number of enterprises and their manufacturing resource are added into the cloud manufacturing platform, manufacturing resource scheduling technology is significant for this platform.

* Corresponding Author

Aiming at the cloud manufacturing resource allocation and scheduling problems, many scholars have done a lot of research. Liu and Zhang [4] proposed a global optimal strategy framework for manufacturing resource service composition in the scenarios of complex multi-task requests, severe QoS constraints on tasks, and services shortage relative to tasks in cloud manufacturing. This optimization strategy of framework was designed considering three aspects of time, cost and reliability, which solve the problem of that the traditional optimization strategy of service composition is inapplicable for complex task oriented manufacturing resource service composition. For solving the problem of manufacturing resource allocation in cloud manufacturing, Su et al. [5] proposed a manufacturing resource service composition assessment index system which included QoS and flexibility factors, and established a bi-level programming model of manufacturing resource allocation on top of the assessment index system. Then fast non-dominated sorting genetic algorithm was used to solve the model.

For achieving optimal selection of heterochthonous manufacturing equipment resources in cloud manufacturing effectively, Yi et al. [6] proposed a manufacturing equipment resource optimal selection model based on shortest time, least cost and highest reliability, and then used an improved genetic algorithm to solve the model. In the above researches related to cloud manufacturing resource allocation and scheduling, the researcher mainly focuses on the coarse-grained manufacturing resources allocation, and does not consider the fine-grained manufacturing resources scheduling problems. The finer the resources granularity, the more complicated it will be in the scheduling process of manufacturing resources. However, the monitoring and control of the state of manufacturing resources will be more accurate, thus more professional cloud manufacturing services can be provided.

Aiming at the "working procedure level" scheduling problems of manufacturing resource (i.e. the fine-grained manufacturing resources scheduling problems) in cloud manufacturing, the scholars have also done some research. Lu et al. [7] proposed the model in which not only the integrated optimization of mixed flow assembly and part processing was considered, but also the collaborative scheduling of cloud service tasks and self-made tasks was considered. Then, a hybrid biogeography optimization algorithm with two level hierarchical structures was proposed to solve the model · Aiming at the utilization of the surplus capacity of manufacturing enterprises in cloud manufacturing, Wang et al. [8] proposed a job shop scheduling model with idle time. An improved second order particle swarm optimization algorithm is adopted to solve the model. The above two models are used to deal with the scheduling problems of fine-grained manufacturing resource from the perspective of the job shops of manufacturing enterprise. So they cannot be used as a resource scheduling model of fine-grained manufacturing for cloud manufacturing platform.

Xiong et al. [9] proposed a multi-objective optimization model for cloud manufacturing scheduling problems, so as to improve the manufacturing efficiency and maintain the load balance. Through an improved particle swarm optimization algorithm combined with simulated annealing algorithm, the rational allocation of complex manufacturing resources can be achieved. For solve the scheduling problem of manufacturing resource in cloud manufacturing, Sun et al. [10] proposed a new multi-objective model to minimize the total time of manufacturing service, the total cost of manufacturing services and the balanced load rate, and then designed a new algorithm which combines genetic algorithm and particle swarm optimization to solve the model. The above two models are used to deal with the scheduling problems of fine-grained manufacturing resource from the perspective of the cloud manufacturing platform. But the former did not consider the character of geographic distribution of manufacturing resource. Although the latter considered the character, when establishing the optimization model, the effect of transportation cost and time on the optimization objectives have still not be considered.

In this paper, from the perspective of the third-party cloud manufacturing service platform, the characteristics of "working procedure level" cloud manufacturing resource scheduling problem are fully analyzed. A scheduling model of manufacturing resources in cloud manufacturing is proposed. This model not only considers the heterochthonous transportation problems that need to be solved when the manufacturing resources located in differing geographical locations are scheduled by cloud manufacturing service platform, but also considers the aspects of service quality and customer satisfaction because that the cloud manufacturing service platform is a special type of service platform. In terms of algorithm, a non-dominated particle swarm optimization algorithm is designed for the multi-objective optimization model. Aiming at the two aspects of working procedure sequencing and resource selection, a double-layer coding is used to design the particle. In the decoding process, the traditional method is improved, in which the waiting time of the working procedure is reduced, and then the processing of the workpiece is speeded up. In order to ensure the diversity and convergence of population, dynamic learning factors are introduced into the evolution process to guide the precise search of the population from large scope to small. The hierarchical selection strategy of non-dominated selection process is optimized, which improves the diversity of population and avoids falling into the local optimal solution. Through the algorithm, the reasonable scheduling and optimization of manufacturing resources are finally achieved, which satisfies the user's multiple optimization objectives for cloud manufacturing resource service.

The rest of the paper is organized as follows. Section 2 outlines the background of the problem and gives the model and constraints of scheduling problem proposed in this paper. Section 3 proposes the improved algorithm, and elaborates its encoding, decoding, evolutionary process, and updating method of optimal solution set, and then gives the process of the improved algorithm. Section 4 gives the corresponding experimental cases. The correctness of the model and the effectiveness of the algorithm are verified through experiments. Section 5 summarizes the paper.

2 Cloud Manufacturing Resource Scheduling Problem

Manufacturing resource scheduling in cloud manufacturing environment is described in Fig. 1. For one cloud manufacturing service request, through the parsing of service request and matching of virtual manufacturing resources in cloud manufacturing platform, there are a number of corresponding cloud manufacturing services that can be provided using the manufacturing resources of different enterprises (i.e. manufacturing resources providers).

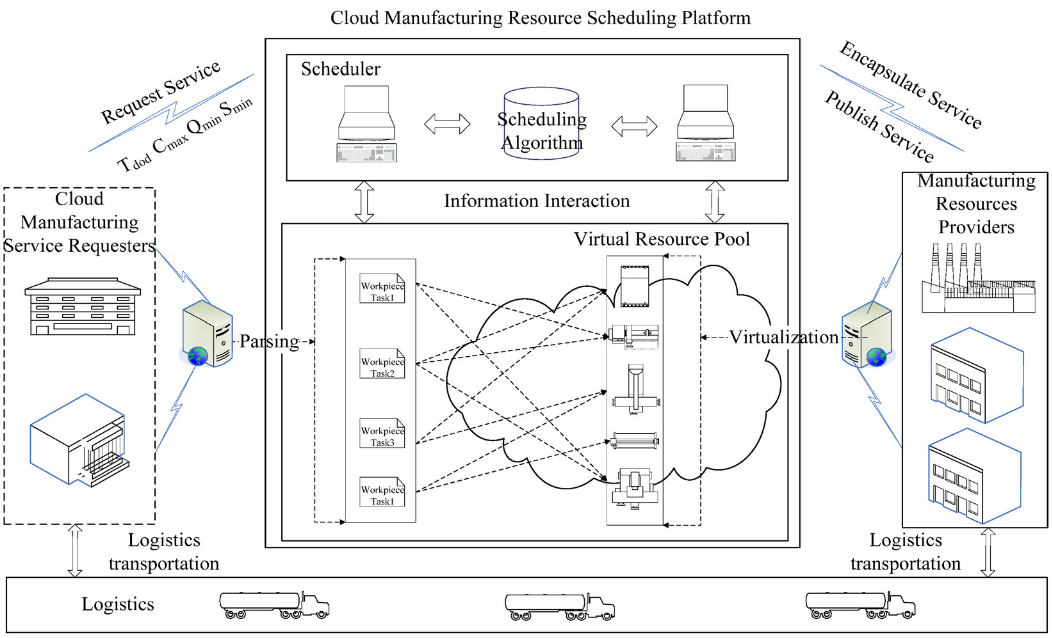

Fig. 1. Manufacturing resource scheduling in cloud manufacturing environment

The service quality of manufacturing resource is objectively evaluated by the cloud manufacturing platform according to service reliability, equipment failure rate and process capability, and the service satisfaction of manufacturing resource would also be subjectively evaluated by the cloud manufacturing service requesters. The cloud manufacturing service requesters can put forward their own personalized requirements in the terms of service completion time, service cost, service quality and service satisfaction. And the scheduler in the platform is responsible for scheduling the manufacturing resource according to the personalized goals of the requesters.

The large amounts of manufacturing resource of different enterprises that are encapsulated and published on the platform are in various geographical areas, and these scattered manufacturing resources make one workpiece-level task (i.e. cloud manufacturing service request) need to go through multiple transportations to complete the all processing. Whenever the scheduler selects the different manufacturing resources for this task, different transportation routes will be formed. In this paper, we will establish a model for the Cloud Manufacturing Resource Scheduling Problem (CMRSP).

Cloud manufacturing resource scheduling problem is given as following: there is one workpiece-level task that consists of n manufacturing tasks (i.e. workpieces), which is represented as a task set $J = \{J_1, J_2,\dots J_n\}$. The jth working procedure of manufacturing task J_i is represented by $O_{i,j}$, i=1, 2, …n and j=1, 2, …l_i, where l_i is the total number of working procedures for manufacturing task J_i. There is also a set of manufacturing resources that are encapsulated as cloud services and then are published on the cloud manufacturing platform. These manufacturing resources are located in multiple different geographical regions, which is represented by $R=\{R_1, R_2,\dots R_r\}$, where r is the total number of the geographical regions, and R_k, k=1, 2, …m is a subset of the manufacturing resources that are located in the kth geographical region. Each task can be processed through various manufacturing resources which may be belonging to different geographical regions.

There are several new characteristics for cloud manufacturing resource scheduling that compared with the traditional resource manufacturing scheduling [8]. Multiple manufacturing resources that are assigned by the scheduler to one cloud service requester may be from different manufacturing enterprises and then located in multiple different geographical regions. So the process of fulfilling one workpiece-level task not only includes the process of processing workpieces, but also includes the transportation of the workpieces that need to continue to be processed on the next manufacturing resources. The transportation factors make the manufacturing resource scheduling become more complicated. According to the scheduling plan for the workpiece-level task, there are the corresponding workpieces transportation routes. The transportation routes are represented as the cloud scheduling transportation matrix P. In the fitness calculation, the P can be used to determine the transportation cost and transportation time, and then improve the efficiency of the algorithm. Matrix P as shown in the following formula (1).

$$P_{r\times r}=\begin{pmatrix} P_{11} & P_{12} & \cdots & P_{1r} \\ P_{21} & P_{22} & \cdots & P_{2r} \\ \cdots & \cdots & \cdots & \cdots \\ P_{r1} & P_{r2} & \cdots & P_{rr} \end{pmatrix}_{r\times r} . \quad \textbf{(1)}$$

Each element of the P is represented by a 3 dimensional vector, $\forall P_{l,k}$=<$subTask$, $Tr\_time_{l,k}$, $Tr\_cost_{l,k}$>, l, k=1,2,…m, where $subTask$ is consisting of a set of procedures that are processed in geographical region l, $Tr\_time_{l,k}$ and $Tr\_cost_{l,k}$ are time and cost of the transportation of those procedures form geographical region l to geographical region k respectively.

Here there is an example of the workpiece-level task as shown in the following formula (2). The manufacturing resources are located in five different geographical regions, so the matrix $P_{5\times 5}$ is given as following. $P_{1,1}$=<$\{O_{1,1}\}$, 0, 0> indicates that procedure $O_{1,1}$ is processed on the manufacturing resource that is located in the 1th geographical region, and before processing $O_{1,1}$, there are not any transportation, so the Tr_time_{11}=0 and Tr_cost_{11}=0.

$P_{3,2}$=<$\{O_{1,3}\}$, 1, 12> indicates that before processing $O_{1,3}$ on the manufacturing resource that is located in the 2th geographical region, the 1th workpiece needs to be transported form the 3th geographical region to the 2th geographical region, and transportation time and cost are 1 and 12 respectively. $P_{1,4}$=<∅, 2, 24> indicates that there are not any workpieces that need to be is transported form the 1th geographical region to the 4th geographical region.

$$P_{5\times 5}=\begin{pmatrix} <\{O_{1,1}\},0,0> & <\{O_{4,3}\},2,24> & <\{O_{1,2}\},1.5,18> & <\varnothing,0,0> & <\varnothing,0,0> \\ <\varnothing,0,0> & <\{O_{5,1}\},0,0> & <\varnothing,0,0> & <\{O_{5,2},O_{5,3}\},2,24> & <\{O_{1,2},O_{4,4}\},1.5,18> \\ <\varnothing,0,0> & <\{O_{1,3}\},1,12> & <\{O_{2,1}\},0,0> & <\varnothing,0,0> & <\{O_{2,2}\},1,12> \\ <\{O_{3,2}\},2,24> & <\varnothing,0,0> & <\varnothing,0,0> & <\{O_{3,1}\},0,0> & <\varnothing,0,0> \\ <\{O_{4,2}\},2.5,30> & <\varnothing,0,0> & <\varnothing,0,0> & <\{O_{2,3}\},1.5,18> & <\{O_{4,1}\},0,0> \end{pmatrix}_{5\times 5} \quad \textbf{(2)}$$

2.1 Mathematical Formulation of the Multidimensional Objective

Considering the characteristics of the cloud manufacturing service, and so as to provide the better service for the requester with the lower cost, the objective of cloud manufacturing resource scheduling problem is designed form the four aspects of service completion time, service cost, service quality and service satisfaction. So the multi-objective optimization is represented as the formula (3).

$$f(X) = (T(X), C(X), Q(X), S(X)) . \tag{3}$$

2.1.1 Service Completion Time

The first sub objective is to minimize the completion time of the cloud manufacturing service for one workpiece-level task, which is represented as the formula (4).

$$T(X) = \min\left(\max_j\left(ET_{i,j}\right)\right),\ ET_{i,j} = ST_{i,j} + t_{i,j,k} . \tag{4}$$

Where $ST_{i,j}$ represents the starting time of processing the working procedure $O_{i,j}$ on the manufacturing resource k. The time $ET_{i,j}$ represents the end time of processing the working procedure $O_{i,j}$ on the manufacturing resource k. The time $t_{i,j,k}$ is the processing time of $O_{i,j}$ on the manufacturing resource k.

2.1.2 Service Cost

The second sub objective is to minimize the total cost of the cloud manufacturing service. The total cost is consisting of processing cost and transportation cost, which is represented as the formula (5).

$$C(X) = \min\left(\sum_{i=1}^{n}\left(\sum_{j=1}^{l_i}\sum_{k=1}^{m}(c_{i,j,k}X_{i,j,k}) + \sum_{j=1}^{l_i} c_{i,j,k}^{tr}\right)\right). \tag{5}$$

Where the cost $c_{i,j,k}$ is the processing cost of the working procedure $O_{i,j}$ on manufacturing resource *k*, and the cost $c_{i,j,k}^{tr}$ is the cost of transporting the *i*th workpiece form the geographical region in which $O_{i,j-1}$ is processed to the geographical region that manufacturing resource *k* is located in. The transportation cost $c_{i,j,k}^{tr}$ can also be obtained from the above matrix *P*. $X_{i,j,k}$ indicates whether the $O_{i,j}$ is processed on the manufacturing resource *k*, if it is, then $X_{i,j,k} = 1$, otherwise $X_{i,j,k} = 0$.

2.1.3 Service Quality

The third sub objective is to maximize the quality of the cloud manufacturing service per unit of processing time. Specifically, this objective is to provide the better manufacturing resources for the requester. Since the better manufacturing resources can be assured that the practical service process is more consistent with the scheduling plan. In cloud manufacturing resource scheduling platform, the manufacturing resources are evaluated according to the evaluation index system that is consisting of three indexes: service reliability, equipment failure rate and process capability. Service reliability means here that the consistency between the practical processing time and the processing time that is promised in the service contract. This sub objective is represented as the formula (6).

$$Q(X) = \max\left(\sum_{i=1}^{n}\sum_{j=1}^{l_i}\sum_{k=1}^{m}\left(Q_k \times t_{i,j,k} \times X_{i,j,k}\right) \Big/ \sum_{i=1}^{n}\sum_{j=1}^{l_i}\sum_{k=1}^{m}\left(t_{i,j,k} \times X_{i,j,k}\right) \right), \tag{6}$$

$$Q_k = \tau_1 q_1 + \tau_2 q_2 + \tau_3 q_3,\ \sum_{i=1}^{3}\tau_i = 1 .$$

Where, the quality Q_k represents the integrated quality of the kth manufacturing resources. It is measured through three indexes. The index q_1, q_2 and q_3 represent the service quality of the kth manufacturing resources in the terms of service reliability, equipment failure rate and process capability. The evaluation criteria of each index are normalized, and their values are belonging to the range [1, 10]. The higher the value, the better the quality.

2.1.4 Service Satisfaction

The fourth sub objective is to maximize the satisfaction of the cloud manufacturing service per unit of processing time. After the platform provide the cloud manufacturing service for the requester, with the help of the provider of manufacturing resources, the requester may evaluate the service. So the large amount of service evaluation historical data was collected, and then on this basis, service satisfaction of each manufacturing resource can be obtained.

$$S(X) = \max \left(\sum_{i=1}^{n} \sum_{j=1}^{l_i} \sum_{k=1}^{m} \left(S_k \times t_{i,j,k} \times X_{i,j,k} \right) \Big/ \sum_{i=1}^{n} \sum_{j=1}^{l_i} \sum_{k=1}^{m} \left(t_{i,j,k} \times X_{i,j,k} \right) \right). \quad \textbf{(7)}$$

Where, the satisfaction S_k is the service satisfaction of the kth manufacturing resource. The value range of [1, 5] is chosen as the evaluation criteria, the higher the value is, the higher the satisfaction is (7).

2.2 Constraint of CMSSP

***Cons* 1:** At a given time, a working procedure can be processed on at most one resource. And then after the current working procedure has been finished, the subsequent working procedure cannot be processed until it obtains the corresponding manufacturing resource. *Cons*1 is represented as the formula (8).

$$ST_{i,j} + t_{i,j,k} + t^{tr}_{i,j+1,h} \le ST_{i,j+1}. \quad \textbf{(8)}$$

Where the time $t^{tr}_{i,j+1,h}$ is the transportation time of $O_{i,j+1}$. Before processing $O_{i,j+1}$, the ith workpiece may be needed to be transported from the geographical region that manufacturing resource k (on which $O_{i,j}$ is processed) is located in to the geographical region that manufacturing resource h is located in. The transportation time $t^{tr}_{i,j+1,h}$ can be obtained from the above matrix P.

***Cons* 2:** At a given time, a manufacturing resource can only processing one working procedure. *Cons* 2 is represented as the formula (9).

$$ST_{i,j} + t_{i,j,k} \le ST_{u,v} \text{ or } ST_{u,v} + t_{u,v,k} \le ST_{i,j}. \quad \textbf{(9)}$$

***Cons* 3**: The maximum value of the end time of all the working procedures is no more than the date of delivery of the workpiece-level task (that is denoted as T_{dod}). *Cons* 3 is represented as the formula (10).

$$\max_{j} \left(ET_{i,j} \right) \le T_{dod}. \quad \textbf{(10)}$$

***Cons* 4:** The total cost (including processing cost and transportation cost) of all the working procedures is no more than the maximum value (that is denoted as $C_{\max}$) of the total service cost that can be accepted by the requester. *Cons* 4 is represented as the formula (11).

$$\sum_{i=1}^{n} \left(\sum_{j=1}^{l_i} \sum_{k=1}^{m} (c_{i,j,k} X_{i,j,k}) + \sum_{j=1}^{l_i} c^{tr}_{i,j,k} \right) \le C_{\max}. \quad \textbf{(11)}$$

***Cons* 5:** The integrated quality (that is denoted as Q) of all the manufacturing resources per unit of processing time is no less than the minimum value (that is denoted as $Q_{\min}$) of the service quality that can be accepted by the requester. *Cons* 5 is represented as the formula (12).

$$Q \geq Q_{\min} . \tag{12}$$

***Cons* 6:** The service satisfaction (that is denoted as S) of all the manufacturing resources per unit of processing time is no less than the minimum value (that is denoted as $S_{\min}$) of the service satisfaction that can be accepted by the requester. *Cons* 6 is represented as the formula (13).

$$S \geq S_{\min} . \tag{13}$$

In the above formula, T_{dod}, $C_{\max}$, $Q_{\min}$ and $S_{\min}$ are all proposed by the requester in the request for the workpiece-level task.

3 Non-dominated Particle Swarm Optimization (NS-PSO)

Particle swarm optimization (PSO) algorithm [9] was proposed by Eberhart and Kennedy in 1995. In the multi-objective evolutionary algorithm, such as NSGA-II [10], PAES [11], SPEA2 [12] have been widely used. This paper proposes an improved PSO algorithm which combines a PSO with non-dominated for cloud manufacturing resource scheduling problem. The NS-PSO algorithm is designed as follows.

First, the non-dominated notion in the particle swarm can be formulated as follows:

In most cases, a multi-objective optimization can be described, without loss of generality, using the following formulation: $\min f(x) = (f_1(x), f_2(x), \cdots, f_M(x))$.

(1) If $\forall 1 \leq q \leq M, f_q(x) \leq f_q(y)$ and at least one index p exists such that $f_p(x) < f_p(y)$, then x *dominates* y (noted $x \succ y$).

(2) If a particle is not dominated by any other particle, then it is non-dominated.

3.1 Coding and Decoding Method

In the cloud manufacturing environment, each manufacturing task (i.e. workpiece) involves multiple working procedures, and for each working procedure, there one manufacturing resource that needs to be selected from multiple candidate manufacturing resources. Therefore, this paper adopts a double-layer particle coding method based on working procedure and resource. The first layer encode is for working procedure and that is denoted as $X_p^t[L]$. t is the current generation. L is the dimension of particle, $L = \sum_{i=1}^{n} l_i$. The same manufacturing task is represented by the same task number, and the working procedure number is represented by the order in which the corresponding task number appears.

For example, the first appearance of the task number 1 is at the second column of the first line (i.e. the first coding layer) in Table 1, which means that the working procedure $O_{1,1}$ is at this column. The second appearance of the task number 1 is at the fifth column means that the working procedure $O_{1,2}$ is at the fifth column. The column number represents the order of processing the working procedures. The position (column number) meaning of the task number 2 and the task number 3 is the same as position meaning of the task number 1.

Table 1. Double-layer particle

procedure	$O_{2,1}$	$O_{1,1}$	$O_{3,1}$	$O_{1,2}$	$O_{3,2}$	$O_{2,2}$	$O_{3,3}$
$X_p^t[L]$	2	1	3	1	3	2	3
$X_m^t[L]$	2	1	3	2	2	3	3

The second layer encode is for resource. The number of the second line (i.e. the second coding layer) in Table 1 is the resource number. For example, the resource 1 appears at the second column of the second line manes that the working procedure that appears at the same column of the first line in table is processed on the resource 1, i.e., the working procedure $O_{1,1}$ is processed on the resource 1.

Due to transportation factors in the cloud manufacturing environment, there are two waiting situations in the resource scheduling process. One is that after the workpiece reaches the processing enterprise, it still needs to wait for the manufacturing resources that are being occupied. Another is that the manufacturing resources are idle, but to it needs to wait for the arrival of the workpiece. Therefore, this paper will decode the particles and calculate the fitness by using the earliest available time decoding algorithm. For each working procedure, it's starting time should be the maximum value between the arrival time of this workpiece and the earliest idle time of it's corresponding manufacturing resource without the effect on the processing process of the other working procedure that is processed on the manufacturing resource. The starting time can also be called as the earliest available time of manufacturing resource.

Pseudo code of decoding algorithm:

(1) For $(q=1, q \le L, q++)$
(2) $O_{i,j} = X_p^t[q]$, $R_k = X_m^t[q]$;
(3) If $O_{i,j}$ is the first working procedure on the R_k;
(4) $ST_{i,j} = \max\{ET_{i,j-1} + t_{i,j,k}^{tr}, 0\} \quad (ET_{i,0} = 0)$;
(5) End If;
(6) Else If R_k have $I(I \ge 1)$ idle time segments;
(7) $S_1 = ET_{i,j-1}(ET_{i,0} = 0)$;
(8) $E_1 = IdleEndTime$;
(9) For $(p=2, p \le I, p++)$
(10) $S_p = nextIdleStartTime$;
(11) $E_p = nextIdleEndTime$;
(12) End For;
(13) If $\exists E_p - S_p \ge t_{i,j,k} + t_{i,j,k}^{tr}$;
(14) $ST_{i,j} = S_{p_{\min}} + t_{i,j,k}^{tr}$;
(15) End If;
(16) Else
(17) $ST_{i,j} = \max\{ET_{i,j-1} + t_{i,j,k}^{tr}, ET_{u,v}\}$;
(18) End Else;
(19) End Else If;
(20) Else
(21) $ST_{i,j} = \max\{ET_{i,j-1} + t_{i,j,k}^{tr}, ET_{u,v}\}$;
(22) End Else;
(23) End For;

In the decoding algorithm, the parameters S_p and E_p represent the start time and end time of the pth idle time segment of manufacturing resource, respectively. The value of transportation time $t_{i,j,k}^{tr}$ can be obtained from the cloud scheduling transportation matrix P (as mentioned in the Section 2). The parameters $O_{i,j}$ and $O_{u,v}$ all represent the working procedure that need to be processed on the resource R_k. The working procedure $O_{i,j}$ can be processed on the resource R_k until $O_{u,v}$ has been finished. The parameter I represents the amount of idle time segments for resource R_k.

For the step 13, when there are multiple idle time segments that can meet the criteria "$E_p - S_p \ge t_{i,j,k} + t_{i,j,k}^{tr}$", the earliest S_p among the start time of these idle time segments should be selected in the step 14, which is represented as $S_{p_{\min}}$.

3.2 Velocity and Position Update

In this paper, on the basis of literature [13], the NS-PSO is proposed. NS-PSO updates the particle through the formula (14):

$$X_i^{t+1} = c_2 \otimes f_3(c_1 \otimes f_2(f_1(X_i^t), p_i^{best})), g_{best}) . \quad \textbf{(14)}$$

X_i^t is the ith particle of the tth generation in the swarm. The particles are updated with the three operators $f_1(X_i^t)$, $f_2(X_i^t)$, $f_3(X_i^t)$. A dynamic learning factor crossover (DLF) is proposed, and the particle velocity is updated by DLF operator.

The process of $f_1(X_i^t)$ operator is given as following.

(1) In the $X_p^t[L]$, the ith element $X_p^t[i]$ is randomly selected, and the positions of the ith element $X_p^t[i]$ and it's next element $X_p^t[i+1]$ are exchanged. At the same time, the positions of the ith element $X_m^t[i]$ and it's next element $X_m^t[i+1]$ need also to be exchanged. If $X_p^t[i] = X_p^t[i+1]$, then the other element $X_p^t[j], j \neq i$ should be randomly selected until $X_p^t[j] \neq X_p^t[j+1]$.

(2) In the $X_m^t[L]$ two element $X_m^t[i]$ and $X_m^t[j]$ are randomly selected, and then for the two working procedures which are processed on the two manufacturing resources $X_m^t[i]$ and $X_m^t[j]$ respectively, two other manufacturing resources should be selected from the their optional manufacturing resources set, and are used to replace the manufacturing resources $X_m^t[i]$ and $X_m^t[j]$ respectively. At the same time the corresponding cloud scheduling transportation matrix P needs also to be update.

The process of $f_2(X_i^t)$ operator is given as following.

NS-PSO algorithm is used to learn personal best and global best particles in the evolution process. In order to avoid the particle swarm fall into a local optimal, the DLF operator is used to adjust the velocity updating in the evolution process of the particles.

Firstly, the cross of working procedure is executed. According to the amount of sub tasks n, the set of sub tasks are divided into two random subsets S_1 and S_2, $|S_1|+|S_2|=n$. For each sub task in S_1, it's corresponding elements in the array $X_p^t[L]$ of X_i^t are copied into the array $newX1_p^{t+1}[L]$ of X_{inew1}^{t+1}, and at the same time the positions of these elements in $newX1_p^{t+1}[L]$ need to be kept consistent with their positions in $X_p^t[L]$. And then for each sub task in S_2, it's corresponding elements in the array $bestX_p^t[L]$ of p_i^{best} are copied into the array $newX1_p^{t+1}[L]$ of X_{inew1}^{t+1} and are used to replace the empty elements in the array $newX1_p^{t+1}[L]$, but the sequence between these elements needs to be kept consistent with their sequence in $bestX_p^t[L]$.

Similarly, for each sub task in S_1, it's corresponding elements in the array $bestX_p^t[L]$ of p_i^{best} are copied into the array $newX2_p^{t+1}[L]$ of X_{inew2}^{t+1}, and at the same time the positions of these elements in $newX2_p^{t+1}[L]$ need to be kept consistent with their positions in $bestX_p^t[L]$. And then for each sub task in S_2, it's corresponding elements in the array $X_p^t[L]$ of X_i^t are copied into the array $newX2_p^{t+1}[L]$ of X_{inew2}^{t+1} and are used to replace the empty elements in the array $newX2_p^{t+1}[L]$, but the sequence between these elements needs to be kept consistent with their sequence in $X_p^t[L]$.

In the above cross of working procedure, the corresponding elements in the array $X_m^t[L]$ and $bestX_m^t[L]$ need to be simultaneously copied into $newX1_m^{t+1}[L]$ and $newX2_m^{t+1}[L]$ whenever the elements in the array $X_p^t[L]$ and $bestX_p^t[L]$ are copied into $newX1_p^{t+1}[L]$ and $newX2_p^{t+1}[L]$. For example, for one element (i.e. working procedure) in the array $X_p^t[L]$, supposing that it's position is j in the array $X_p^t[L]$, when this element is copied from $X_p^t[L]$ into $newX1_p^{t+1}[L]$, it's corresponding element $X_m^t[j]$ in the array $X_m^t[L]$ needs to be copied simultaneously from $X_m^t[L]$ into $newX1_m^{t+1}[L]$.

Secondly, cross of resources is executed. An array $R[L]$ are randomly generated, each element in $R[L]$ is a probability value, and it's range of value is (0, 1). The ith probability value in $R[L]$ represents the possibility of that the resources $newX1_m^{t+1}[i]$ of X_{inew1}^{t+1} and $newX2_m^{t+1}[i]$ of X_{inew2}^{t+1} are replaced. For each probability value in R [L], it is compared with the factor $c(t)$, and if R [i] is less than $c(t)$, then the resources $newX1_m^{t+1}[i]$ and $newX2_m^{t+1}[i]$ are replaced with $bestX_m^t[i]$ of p_i^{best}. At last the fitness of X_{inew1}^{t+1} and X_{inew2}^{t+1} are calculated by using the above decoding algorithm. So the non-dominated particle is identified from two particles X_{inew1}^{t+1} and X_{inew2}^{t+1}, and then is taken as the next generation of particle X_i^{t+1}.

In the formula (15), $c(t)$ is assigned the larger value in the beginning phase of algorithm iteration, which is convenient for particles to expand the search space. This makes sure that the working procedures are able to combine adequately with different resources, so as to avoid falling into local optimum. With the increasing of iterations, the value of $c(t)$ becomes smaller, so the local search capability of particles is enhanced, which are helpful converge quickly to the optimal solution.

$$c(t) = c_{start} - (c_{start} - c_{end})(t / T_{\max})^2 . \quad \textbf{(15)}$$

Where $c_{start} = 0.9, c_{end} = 0.2$, t is the current generation, $T_{\max}$ the maximum generation.

The process of $f_3(X_i^t)$ operator is the same as the one of $f_2(X_i^t)$ operator. A non-dominated particle is randomly selected from the set P_t^{gbest}, and is denoted as g_{best}. For $f_3(X_i^t)$ operator, the role of g_{best} is the same as the one of p_i^{best} in $f_2(X_i^t)$ operator.

3.3 Update Goal Best Solution Set

The update process of global optimal solution set is shown in Fig. 2. Through evolution, the set P_{t-1}^{gbest} of the goal best solution g_{best} is transformed into a new solution set P_t^{gbest}. The set P_{t-1}^{gbest} and P_t^{gbest} are merged and copied into the set R_t^{gbest}. The solutions of set R_t^{gbest} are sorted by their non-dominated relationship between them, and then the crowding distances between these solutions are calculated. During selecting particles from the set R_t^{gbest} and inserting back them into the set P_t^{gbest}, the selection rules are as follows.

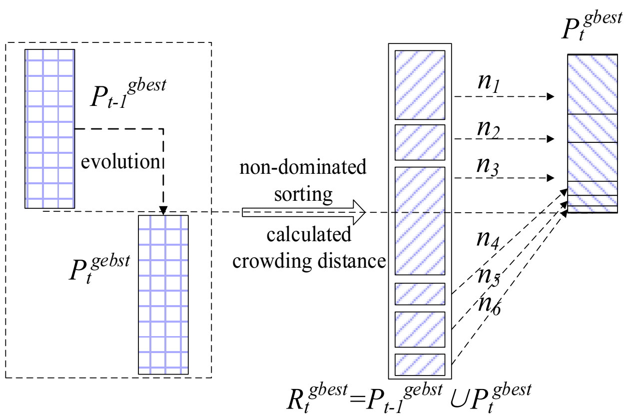

Fig. 2. Update process of goal best solution set

***Rules* 1.** The particle with the higher non-domination rank is constantly being selected in the first place.
***Rules* 2.** If there are several particles with the same non-domination rank through **Rules 1**, then the particle with the larger crowding distance being selected in the first place.

In addition, the formula (16) of the dynamic adjustor is also used in the process of selecting particles.

$$n_i = \lfloor |F_i| / i \rfloor . \quad \textbf{(16)}$$

Where, i denotes the ith front, i.e., the ith non-domination rank, and $|F_i|$ denotes the number of particles in the ith front. n_i denotes the number of particles that should be selected from the ith front.

In the traditional process of selecting particles according to their non-dominated relationship, the particles with the higher non-domination rank are taken as the better choices, but several of these better choices may be far away from the true optimal solution. By using the formula (16), several particles with the lower non-domination rank might also be selected to increase the diversity of the swarm. At last, according to the above rules and the formula (16), the N particles are selected and inserted back into the set P_t^{gbest} .

3.4 NS-PSO Steps

```
(1)  initializeParticlesSwarm();
(2)  t =1, T_max =300 , P_0^gbest =∅ ;
(3)    while t ≤ T_max  do
(4)  for each particle do;
(5)  calculateFitness();      // Formulas (4), (5), (6), (7)
(6)  updatePbest();           // by non-dominated sorting
(7)  updatePosition();        // Formulas (14)
(8)  end for
(9)  updateGbestSet();        // Rule1,2 and Formulas (16)
(10) t++;
(11)   end while
(12) returnGbestSet();
```

4 Example and Algorithm Analysis

The algorithm is programmed by MATLAB 2014, and it's operating environment is the second-generation Intel Core i5-2430M processor with 4G memory. In the section 4.1 the above algorithm is tested by an example and the results are compared with SPEA2 and NSGA II.

4.1 Test and Analysis

In order to verify the capability of algorithm to deal with the real-life problems, an example of cloud manufacturing resource scheduling problem from a certain electrical machinery plant is given here. In this example, the actual production data is preprocessed and then taken as the original experimental data of cloud manufacturing resource scheduling problem, as shown in the following three tables.

Table 2. Example of cloud manufacturing resource scheduling

Tasks	Procedures	Resource	Service Time
J_1	$O_{1,1}$	$R_3/R_5/R_7$	3/3/5
	$O_{1,2}$	$R_1/R_6/R_7$	5/8/8
	$O_{1,3}$	R_2/R_5	6/3
	$O_{1,4}$	R_5/R_9	4/3
J_2	$O_{2,1}$	R_4/R_9	10/4
	$O_{2,2}$	$R_2/R_5/R_{10}$	8/5/4
	$O_{2,3}$	R_3/R_8	3/5
	$O_{2,4}$	$R_1/R_3/R_7$	6/2/6
J_3	$O_{3,1}$	$R_3/R_9/R_{10}$	4/3/4
	$O_{3,2}$	$R_1/R_5/R_{10}$	3/4/3
	$O_{3,3}$	$R_3/R_4/R_9$	3/6/4
J_4	$O_{4,1}$	$R_2/R_4/R_7$	6/8/7
	$O_{4,2}$	$R_1/R_6/R_8$	5/8/7
	$O_{4,3}$	$R_1/R_4/R_{10}$	5/9/3
J_5	$O_{5,1}$	$R_5/R_6/R_9$	2/6/2
	$O_{5,2}$	R_3/R_4	3/7
	$O_{5,3}$	$R_2/R_7/R_{10}$	5/4/3
J_6	$O_{6,1}$	$R_1/R_3/R_5/R_8$	4/3/4/6
	$O_{6,2}$	R_2/R_6	5/6
	$O_{6,3}$	$R_4/R_5/R_{10}$	6/4/6
	$O_{6,4}$	$R_2/R_6/R_9$	8/10/6

Table 2. Example of cloud manufacturing resource scheduling (continue)

Tasks	Procedures	Resource	Service Time
J_7	$O_{7,1}$	$R_4/R_7/R_{10}$	10/7/5
	$O_{7,2}$	R_3/R_6	3/8
	$O_{7,3}$	R_2/R_8	5/5
J_8	$O_{8,1}$	R_3/R_8	4/7
	$O_{8,2}$	$R_2/R_6/R_9$	3/7/2
	$O_{8,3}$	$R_2/R_4/R_8$	5/4/6
	$O_{8,4}$	$R_1/R_6/R_{10}$	5/6/6
J_9	$O_{9,1}$	$R_1/R_5/R_7$	3/3/9
	$O_{9,2}$	$R_3/R_6/R_9$	3/6/4
J_{10}	$O_{10,1}$	$R_2/R_3/R_8$	5/3/4
	$O_{10,2}$	$R_1/R_5/R_8$	5/3/4
	$O_{10,3}$	$R_1/R_9/R_{10}$	5/4/4
$Q_{\min}$=9.60, $S_{\min}$=4.70			

Table 3. Service cost per unit time, service quality, satisfaction of manufacturing resources

Resource	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}
Service Cost of unit time	16	12	21	10	18	11	15	13	20	19
Service Quality	9.7	9.6	9.8	9.5	9.7	9.5	9.7	9.6	9.8	9.8
Satisfaction	4.8	4.7	4.9	4.5	4.9	4.6	4.7	4.7	4.9	4.8

Table 4. Transportation time, transportation cost per unit time of manufacturing resources

Transportation Time	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}
R_1	0	2	1	2	2	1	2.5	1	2.5	2.5
R_2	2	0	1	2	2	1	2.5	2.5	1	1
R_3	1	1	0	1	1	1.5	1.5	1.5	1.5	2
R_4	2	2	1	0	2	2.5	1	1	2.5	3
R_5	2	2	1	2	0	2.5	1	2.5	1	2.5
R_6	1	1	1.5	2.5	2.5	0	3	2	2	1.5
R_7	2.5	2.5	1.5	1	1	3	0	2	2	3.5
R_8	1	2.5	1.5	1	2.5	2	2	0	3	3.5
R_9	2.5	1	1.5	2.5	1	2	2	3	0	1.5
R_{10}	2.5	1	2	3	2.5	1.5	3.5	3.5	1.5	0
The transportation cost per unit time: 12										

Three algorithms, SPEA2, NSGA-II and NS-PSO, are run 20 times, each loop 300 generations. The result overall value estimation method [14] is used to select the scheduling plan. Table 5 shows the results the overall value of the highest of the example. It can be seen from the results that the NS-PSO algorithm is superior to the other two algorithms in terms of service time and quality, and NS-PSO's scheduling plan overall value is better than the other two algorithms.

Table 5. Example result comparison

Objective	SPEA2	NSGA-II	NS-PSO
Time	27	27	25
Cost	2880	2870	2872
Quality	9.63	9.63	9.65
Satisfaction	4.74	4.70	4.72

Fig. 3 shows the convergence of the service time of different algorithms. It can be seen from Fig. 3 that the NS-PSO algorithm performs wider search space by using the DLF factor in the early stage. The convergence results after the 150th generation are better than the other two algorithms. In the later stage, the position is updated in a smaller space, so that the algorithm can converge to the optimal solution. It can be seen from Fig. 3 that the service time 25 is better than the other two algorithms.

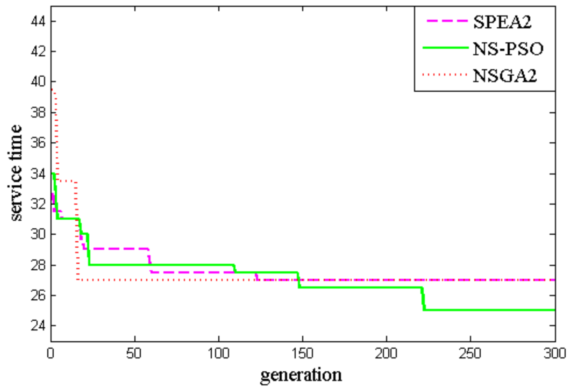

Fig. 3. The convergence graph of task completion time

Fig. 4 shows the proportion of non-dominated solutions during the calculation process. It can be seen that the convergence speed of the algorithms are close, but the proportion of non-dominated solution of NS-PSO is higher than the other two algorithms, which improves the search efficiency. It can provide higher valuable solutions of better scheduling plan to users. Based on the solution set of calculations, the optimal scheduling plan is shown in Fig. 5.

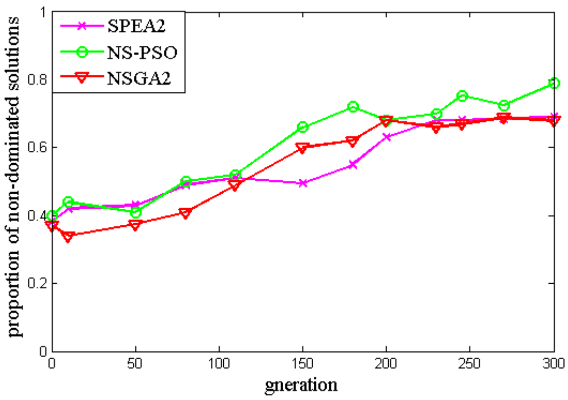

Fig. 4. Proportion of non-dominated solutions

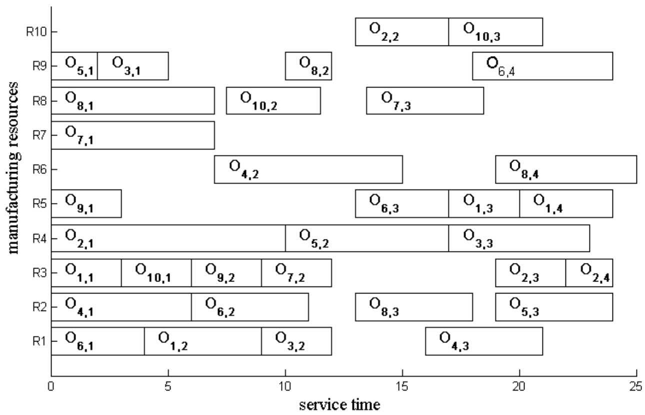

Fig. 5. The Gantt chart of optimal solution

5 Conclusions

For cloud manufacturing resources scheduling problem, this paper proposed the multi-objective scheduling optimization model based on the objectives: minimizing the makespan and the total cost, maximizing the quality of service and the users' satisfaction. An improved particle swarm optimization algorithm, abbreviated as NS-PSO, is proposed to solve this model. By discretizing the traditional PSO algorithm that is used for solving cloud manufacturing resource scheduling problem, and combining with pareto optimization, a set of optional optimization solutions can be obtained through one calculation. The NS-PSO algorithm can converge to the optimization solution by using the parallelism of the particle swarm and the optimization particles-oriented character. The swarm is updated by using the improved non-dominated sorting method, so the overall value of the solution set is improved, which can bring more valuable reference for selecting the scheduling plan. Finally, an example is used to verify the effectiveness of the model and algorithm in the real-life production environment. And the performance of the algorithm is proved by the test and analysis of the benchmark problem. The cloud manufacturing scheduling problem will be a focus of future research on scheduling problem. The proposed problem model and scheduling algorithm can provide an effective theoretical basis for future research on cloud manufacturing scheduling problem.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 51375128) and University Nursing Program for Young Scholars with Creative Talents in Heilongjiang Province (UNPYSCT-2016032).

References

[1] F. Tao, Y. Cheng, L.-D Xu, CCIoT-CMfg: Cloud computing and internet of things-based cloud manufacturing service system, IEEE Transactions on Industrial Informatics 10(2)(2014) 1435-1442.

[2] H. Huang, Z. Wang, Y.-J. Ji, Y. Yan, Analysis on distributed networked cloud manufacturing mode, Modern Manufacturing Engineering (11)(2017) 36-42.

[3] B.-H. Li, X.-D. Cai, L. Zhang, Smart cloud manufacturing: a new kind of manufacturing paradigm, approach and ecosystem of deep integration of the internet and the manufacturing industry, ZTE Technology 22(5)(2016) 2-6.

[4] B. Liu, Z.-L. Zhang, Framework of complex task oriented service composition and optimization in cloud manufacturing systems, China Mechanical Engineering 26(8)(2015) 1048-1057.

[5] K.-K. Su, W.-S. Xu, J.-Y. Li, Manufacturing resource allocation method based on bi-level programming in cloud manufacturing, Computer Integrated Manufacturing Systems 21(7)(2015) 1941-1952.

[6] A.-B Yi, X.-F. Yao, H.-P Zhou, C.-J. Zhang, Multi-objective optimal selection of equipment resources in cloud manufacturing, Computer Integrated Manufacturing Systems 23(6)(2017) 1187-1195.

[7] J.-X. Lu, Q.-H. Hu, Q.-Y. Dong, H.-T. Tang, Cloud manufacturing-oriented mixed-model hybrid shop-scheduling problem, China Mechanical Engineering 28(2)(2017) 191-198.

[8] Z. Wang, J.-H. Zhang, Y.-Q. Qi, Job shop scheduling method with idle time in cloud manufacturing, Control and Decision 32(5)(2017) 811-816.

[9] Y.-H. Xiong, J. Wang, M. Wu, J.-H. Yu, Virtual resource scheduling for cloud manufacturing based on multi-objective optimization, Computer Integrated Manufacturing Systems 21(11)(2015) 3079-3087.

[10] W.-H. Sun, H.-Y. Wu, W.-X. Lv, Y.-C. Gao, Research on multi-objective process scheduling of cloud manufacturing resources, Journal of Nanjing University of Aeronautics & Astronautics 49(6)(2017) 773-778.

[11] G. Xiao, X.-D. Ke, Y.-M. Zhang, J.-W. Lu, Z.-J. Zhang, Research on application mode and key technology of cloud manufacturing for industry alliance 46(1)(2018) 11-20.

[12] Coello C A C, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization, IEEE Transactions on Evolutionary Computation 8(3)(2004) 256-279.

[13] Deb K, Pratap A, Agarwal S, T Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6(2)(2002) 182-197.

[14] Knowles J D, Corne D W, M-PAES: a memetic algorithm for multi-objective optimization, in: Proc. the 2000 Congress on Evolutionary Computation, 2000.