

## Research Article

# A New Manufacturing Service Selection and Composition Method Using Improved Flower Pollination Algorithm

Wenyu Zhang, Yushu Yang, Shuai Zhang, Dejian Yu, and Yangbing Xu

School of Information, Zhejiang University of Finance and Economics, Hangzhou 310018, China

Correspondence should be addressed to Shuai Zhang; [zs760914@sina.com](mailto:zs760914@sina.com)

Received 12 August 2016; Revised 16 November 2016; Accepted 22 November 2016

Academic Editor: Eusebio Valero

Copyright © 2016 Wenyu Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With an increasing number of manufacturing services, the means by which to select and compose these manufacturing services have become a challenging problem. It can be regarded as a multiobjective optimization problem that involves a variety of conflicting quality of service (QoS) attributes. In this study, a multiobjective optimization model of manufacturing service composition is presented that is based on QoS and an environmental index. Next, the skyline operator is applied to reduce the solution space. And then a new method called improved Flower Pollination Algorithm (FPA) is proposed for solving the problem of manufacturing service selection and composition. The improved FPA enhances the performance of basic FPA by combining the latter with crossover and mutation operators of the Differential Evolution (DE) algorithm. Finally, a case study is conducted to compare the proposed method with other evolutionary algorithms, including the Genetic Algorithm, DE, basic FPA, and extended FPA. The experimental results reveal that the proposed method performs best at solving the problem of manufacturing service selection and composition.

## 1. Introduction

With the rapid development of the Internet and associated information technology, many traditional manufacturing industries have changed the ways by which they provide products and services. Suppliers increasingly encapsulate their production capacities into services and upload them to shared network platforms. In general, user requirements are extremely complex and cannot be satisfied by a single manufacturing service. Therefore, to meet such increasingly complex requirements, appropriate candidate services must be chosen from a service set for each subtask and combined to form a composite service. The optimal composite manufacturing service is then identified from all the possible solutions and used to perform the multiobjective task [1]. As an increasing number of manufacturing services become available, the manner in which to select and compose them appropriately is a challenging problem. This is referred to as a multiobjective manufacturing grid resource service composition and optimal-selection problem [1].

As for the background of manufacturing service selection and composition, the concept of cloud manufacturing, as a new manufacturing paradigm, was proposed by Zhang et al.

[2] and has attracted considerable attention from experts and scholars in the field of manufacturing service. Tao et al. [3] presented the classification of manufacturing resource and service and proposed a five-layered structure (i.e., resource layer, perception layer, network layer, service layer, and application layer) toward cloud manufacturing.

With the rapid increase in the number of web services having the same functional attributes, selecting an appropriate candidate service based on nonfunctional attributes such as quality of service (QoS) has become an equally challenging task that has also attracted considerable attention [4]. As with web services, each manufacturing service has functional and nonfunctional attributes [5]. Regarding nonfunctional attributes, QoS has become crucial in evaluating the overall performance of manufacturing service composition. Time, cost, reliability, and availability are four representative QoS properties that are frequently employed by researchers. Tao et al. [6] presented a QoS description supporting resource service correlation. They also described the basic resource service composite modes (RSCM) for composite resource service and the methods to translate a complicated RSCM into a simple sequence RSCM.

However, it is worth noting that, unlike typical web service composition, additional restrictions on manufacturing service composition exist [7]. For example, manufacturing enterprises belong to the entity enterprises that often produce considerable amount of contamination in the course of providing services. Therefore, from the perspective of green manufacturing [7], an environmental index (i.e., carbon emission) must be factored into the evaluation system to measure the overall performance of a manufacturing service composition. In addition, as the number of candidate services increases, the solution space becomes so large that finding the optimal solution is difficult. Thus, some researchers have proposed using the skyline operator [8] for service composition and have advanced the concept of skyline service [9, 10] to limit the size of the solution space and improve the combined utility of composite solutions.

Even after applying of the skyline operator, selecting an optimal composite solution remains time-consuming. Although many evolutionary algorithms, like Genetic Algorithm (GA), Particle Swarm Optimization (PSO) algorithm, and Differential Evolution (DE) algorithm, have been proposed to solve this problem, room remains for improvement. The Flower Pollination Algorithm (FPA) was first proposed by Yang [11] in 2012 as a new metaheuristic algorithm. It has been applied to many fields and has been shown to perform well, especially at solving optimization problems.

In this study, a novel and efficient method is proposed that employs not only the skyline operator but also the improved FPA (IFPA) to solve the problem of QoS-based manufacturing service selection and composition. The use of the skyline operator means that only those services that belong to the skyline service set are valid candidate services [9]. In this study, the differences between our proposed IFPA and basic FPA are the following three points. First, a grouping strategy and elite replacement strategy are used to improve the convergence speed of FPA. Second, we not only use the adaptive parameter to adjust dynamically the switch probability between global and local pollination but also employ an adaptive scaling factor to control the step size of the global search. Third, we introduce crossover and mutation operators of the DE algorithm into the local search of FPA to improve its performance.

The rest of this paper is organized as follows. Related studies about manufacturing service composition and the development of relevant evolutionary algorithms are discussed in Section 2. Section 3 provides a description of the model that is established in this study and definitions of skyline service. The basic FPA and its improvements are introduced in Section 4. Section 5 offers a case study and several comparative experiments used to illustrate the practicality and effectiveness of the proposed method. A conclusion that outlines the authors' main contributions and that describes future research is provided in Section 6.

## 2. Related Work

*2.1. Manufacturing Service Selection and Composition.* With the rapid development of cloud computing and the advent

of big data, the problem of optimizing composite services is causing increasing concern in the manufacturing industry. A composite service is the result of aggregating multiple services into one, and its aim is to execute functions of greater complexity than those that can be performed by a single service [12].

To evaluate the overall performance of service composition, Zeng et al. [13] set QoS attributes as the main measurement, including execution price, execution time, reputation, reliability, and availability, which were widely accepted and applied. To obtain high quality of service composition, they also introduced a middleware platform to express user satisfaction as utility functions. Tao et al. [14] presented different aggregation formulations for four different structures of service composition, including sequence, parallel, selective and cycle. In addition, the correlation of service composition was taken into consideration. To minimize implementation time and cost and to maximize reliability, an improved Particle Swarm Optimization (PSO) algorithm was introduced as the key technology. Considering the heterogeneous QoS values and homogeneous resources in manufacturing service, Zheng et al. [15] took the dynamic customer requirements into account by introducing the fuzzy theory and presented the way of calculating fuzzy QoS value and selecting the best services considering design preference. Hao et al. [16] pointed out that, in order to obtain the best service composition solution, user QoS constraints need to be relaxed rather than overlooked, so they proposed a novel QoS model and computational framework. Tao et al. [17] made an overview about manufacturing service management and analyzed the future research directions of manufacturing service composition and optimization.

However, most of the aforementioned studies only focus on the general QoS attributes and ignore the importance of environmental protection, which has become one of the most important attributes in the process of selecting manufacturing service [7, 17]. Therefore, an environmental index such as carbon emission is added in this paper as a soft constraint to measure the overall performance of manufacturing service composition.

As a multiobjective optimization problem, its solution space is very large and the objectives of manufacturing service composition are in conflict with each other. Thus, Alrifai et al. [18] first introduced the skyline operator [8] into service composition to reduce the number of candidate services. They also proposed the definition of skyline service to improve the efficiency of service selection. Skyline service has been widely used by scholars and become an active research area [10, 12]. In view of its good performance, the proposed method in this paper incorporates the skyline service concept in the model to limit the size of the solution space and improve the overall quality of the optimal manufacturing service composition solution.

*2.2. The Development of Evolutionary Algorithms in Manufacturing Service Composition.* In order to solve the problem of service composition, experts and scholars have used a variety of evolutionary algorithms. For example, in the authors'

previous work [19], the extended GA was employed to solve the problem of manufacturing resource composition, treating the overall utility of multiple objectives as the metric of the composite solution. Li et al. [4] employed a multipopulation GA to solve the problem of QoS-oriented web service composition for the Internet of Things. Tao et al. [1] presented the concept of multiobjective manufacturing grid resource service composition and optimization problem and used the PSO algorithm to solve it. They not only introduced the basic resource service composite modes (RSCM) and the principles to translate a complicated RSCM into a simple sequence but also used the extended PSO algorithm to improve the quality of the optimal solution. Pop et al. [20] introduced a modified DE algorithm with a new genome-encoding method to solve the problem of service composition. Wei and Liu [21] introduced the Ant Colony Optimization (ACO) algorithm into the cloud manufacturing resource allocation model. Wang et al. [22] proposed a new optimization algorithm called the culture max-min ant system, which not only established a novel comprehensive model including both generic and domain QoS but also combined the Culture Algorithm (CA) and the ACO algorithm into a new approach. Lartigau et al. [7] applied an adapted Artificial Bee Colony (ABC) optimization algorithm to the cloud manufacturing service composition model, which not only included QoS attributes but also took the geoperspective transportation into consideration.

Although many conventional evolutionary algorithms have been used in service composition, each has its own deficiencies, for example, converging too easily to local optimum or premature convergence. In order to obtain better optimization solutions, a variety of emerging algorithms have been proposed. For example, inspired by the biological process of flower pollination, Yang [11] first proposed FPA in 2012 and proved that it was more efficient in solving optimization problems than both the GA and PSO.

As a novel and effective metaheuristic algorithm, FPA has been used in many fields. Abdelaziz and Ali [23] used it in a multimachine power system to seek the optimal parameters of static VAR compensator damping controller. Abdelaziz et al. [24] also used FPA to solve the problem of economic load dispatch and combined economic emission dispatch problems. To solve sizing optimization problem of truss structures, Bekdaş et al. [25] used FPA to minimize the weight of truss structures. To obtain feasible optimized designs, they added an iterative constraint handling strategy into the FPA search engine. El-henawy and Ismail [26] extended the basic FPA by combining it with chaotic search to solve large integer programming problems. Wang et al. [27] applied FPA with bee pollinator in the field of cluster analysis to make up for deficiencies in the  $k$ -means method. By comparing it with other algorithms such as GA, they proved the better accuracy and higher level of stability of their proposed method. In order to solve the problem of multilevel image thresholding, Ouadfel and Taleb-Ahmed [28] applied FPA to search for the best values from a given number of thresholds with the purpose of maximizing the objective function. To apply the FPA to solve the antenna positioning problem, Dahi et al. [29] proposed four binary variants of the FPA by applying

the principal mapping techniques. In addition, to study the applicability of this new metaheuristic algorithm in real-world applications, Draa [30] reviewed the performance of FPA in terms of both qualitative and quantitative aspects, and then he summarized the defects of the basic FPA by comparing it with state-of-the-art algorithms and pointed out the extension and application directions of FPA.

However, even though FPA research is expanding, to the best of our knowledge, it has seldom been used to solve the optimization problem of service composition, let alone be applied in the field of manufacturing service selection and composition. Therefore, in this study, FPA is applied to solve the manufacturing service selection and composition problem by improving the standard FPA in combination with DE. The experimental results show that the proposed IFPA can obtain better optimal solutions than other algorithms such as GA [4], DE [31], basic FPA [11], and even extended FPA (EFPA) [32].

### 3. Model of Manufacturing Service Composition Problem

#### 3.1. Notations in Manufacturing Service Composition Model.

To solve the manufacturing service composition problem and translate it into a mathematical model, we use the following notations to better illustrate the model of manufacturing service composition:

$T$ : execution time of the candidate service

$C$ : execution cost of the candidate service

$Ava$ : availability of the candidate service

$Rel$ : reliability of the candidate service

$m$ : the number of subtasks in each basic structure

$MS_i$ : the selected manufacturing service realizing  $i$ th subtask,  $i = 1, 2, \dots, m$

$T(seq)$ : the total execution time in the sequence structure

$C(seq)$ : the total execution cost in the sequence structure

$Ava(seq)$ : the total availability in the sequence structure

$Rel(seq)$ : the total reliability in the sequence structure

$Ec(MS_i)$ : the processed carbon emission value of  $i$ th manufacturing service

$Ec^*$ : a soft constraint about carbon emission that is provided by users

$\theta_i$ : real carbon emission value of  $i$ th candidate service.

3.2. *The Calculation of QoS Properties.* In this study, the four representative QoS properties of execution cost ( $C$ ), execution time ( $T$ ), reliability ( $Rel$ ), and availability ( $Ava$ ) are employed as criteria for manufacturing service performance evaluation [13, 33].

As there are various metrics for different QoS attributes, in order to ensure the accuracy of assessment results, all the

QoS attributes need to be normalized and converted into the same order of magnitude before they are employed in the final QoS aggregation formula. It has been well recognized that service quality is negatively correlated with cost and time, but it has a positive correlation with reliability and availability. Therefore, each QoS attribute is classified as being either positive attribute or negative attribute. The corresponding calculation formulas are as follows [5]:

$$q_{\text{nor}} = \begin{cases} \frac{q - q_{\min}}{q_{\max} - q_{\min}} & \text{if } q_{\max} \neq q_{\min} \\ 1 & \text{if } q_{\max} = q_{\min} \end{cases} \quad (1)$$

positive attribute,

$$q_{\text{nor}} = \begin{cases} \frac{q_{\max} - q}{q_{\max} - q_{\min}} & \text{if } q_{\max} \neq q_{\min} \\ 1 & \text{if } q_{\max} = q_{\min} \end{cases} \quad (2)$$

negative attribute.

Here,  $q$  is  $k$ th true attribute value of a manufacturing service and  $q_{\text{nor}}$  represents the corresponding normalized value.  $q_{\max}$  represents the maximal value of  $k$ th QoS attribute for manufacturing services in the same candidate service set, that is,  $q_{\max} = \max\{q_1, q_2, \dots\}$ ;  $q_{\min}$  stands for the minimal value of  $k$ th QoS attribute for manufacturing service, that is,  $q_{\min} = \min\{q_1, q_2, \dots\}$ .

In general, manufacturing service composition includes four basic structures: sequence, parallel, selective, and cycle structures. There are different aggregation methods for each structure. The aggregation formulas are defined in [13]. For example, the following are the calculation formulas for the sequence structure:

$$\begin{aligned} T(\text{seq}) &= \sum_{i=1}^m T(\text{MS}_i), \\ C(\text{seq}) &= \sum_{i=1}^m C(\text{MS}_i), \\ \text{Rel}(\text{seq}) &= \prod_{i=1}^m \text{Rel}(\text{MS}_i), \\ \text{Ava}(\text{seq}) &= \prod_{i=1}^m \text{Ava}(\text{MS}_i). \end{aligned} \quad (3)$$

In order to use a unified standard to measure the overall performance of manufacturing service composition, the multiple objectives need to be transformed into a single one by assigning different weights to each attribute according to the different preferences of users. The final QoS evaluation value of manufacturing service composition can be calculated according to the following equation:

$$f(\text{QoS}) = w_1 T + w_2 C + w_3 \text{Rel} + w_4 \text{Ava}, \quad (4)$$

where  $w_1, w_2, w_3$ , and  $w_4$  sum to 1 and represent the respective weights of the four attributes, including total time ( $T$ ), total cost ( $C$ ), total reliability ( $\text{Rel}$ ), and total availability ( $\text{Ava}$ ), of manufacturing service composition.

**3.3. The Calculation of Environmental Index.** In addition to the basic QoS indicators, the proposed model contains an environmental indicator. Customers generally consider the QoS performance of manufacturing service composition to be more important than its environmental performance, and therefore they have different expectations of them. For that reason, the constraints proposed by customers can be divided into two types: rigid constraints and soft constraints. QoS constraints are defined as being rigid ones that must be satisfied, whereas environmental indices are defined as soft constraints that do not have to be satisfied strictly [22].

In the present paper, the authors regard carbon emission as an environmental protection index. As with manufacturing tasks, customers impose their own soft constraints that should be satisfied as far as possible. For each candidate manufacturing service  $\text{MS}_i$ , the deviation value between its carbon emission and soft constraint is calculated according to the following equation [22]:

$$\begin{aligned} \text{Ec}(\text{MS}_i) &= \frac{\theta_i - \text{Ec}^*}{\text{Ec}^*} \times \beta, \\ \beta &= \begin{cases} 0, & \text{if } \theta_i \text{ satisfy } \text{Ec}^* \\ 1, & \text{if } \theta_i \text{ do not satisfy } \text{Ec}^* \end{cases} \end{aligned} \quad (5)$$

where  $\text{Ec}(\text{MS}_i)$  is the deviation carbon emission value,  $\theta_i$  is the real value (i.e., carbon emission value) provided by suppliers, and  $\text{Ec}^*$  is a soft constraint about carbon emission that is provided by users. From the above equation, it is clear that  $\beta$  is a binary index. If carbon emission meets the user requirement for environmental protection, then  $\beta = 0$ ; otherwise,  $\beta = 1$ . It is well known that the smaller the deviation value, the higher the environmental indicator. Thus, the deviation value is a negative attribute, and its standardized formula is defined in (2) too. The aggregation formulas of deviation value about carbon emission are similar to the aggregation formula of execution cost, so we omitted them here.

**3.4. The Comprehensive Utility of a Manufacturing Service Composition.** Based on the above description, the final objective function of manufacturing service composition is defined as

$$\max F(\text{MSC}) = \lambda_1 f(\text{QoS}) + \lambda_2 f(\text{Ec}), \quad (6)$$

where  $F(\text{MSC})$  is the comprehensive utility of a manufacturing service composition,  $f(\text{QoS})$  represents the final QoS evaluation, and  $f(\text{Ec})$  represents the final carbon emission value of manufacturing service composition. Terms  $\lambda_1$  and  $\lambda_2$  sum to 1 and represent the weights of QoS and environmental index in a manufacturing service composition solution, respectively. In this model, the goal is to obtain the maximum value of the comprehensive utility, and this comprehensive utility be regarded as the fitness value in the experiment section of this paper.

3.5. *The Introduction of Skyline Service.* Before the new evolutionary algorithm is carried out to solve the problem of manufacturing service composition, the skyline service is introduced to reduce the number of candidate manufacturing services and improve the quality of optimal solution. The concept of skyline service, which was based on dominance relationship, was first proposed by Alrifai et al. [18]. The following is a detailed description of the related definitions [9].

*Definition 1 (dominance).* One assumes that there are two vectors  $X_1 = (x_{11}, x_{12}, \dots, x_{1d})$  and  $X_2 = (x_{21}, x_{22}, \dots, x_{2d})$  in  $d$ -dimensional space. If the following two conditions are satisfied, one says that  $X_1$  dominates  $X_2$ . Firstly, each dimension in  $X_1$  is greater than or equal to the corresponding in  $X_2$ . Secondly, at least one dimension in  $X_1$  is strictly greater than the corresponding in  $X_2$ . The two conditions can be expressed mathematically as follows [34]:

$$\begin{aligned} \text{if } \forall i \in \{1, 2, \dots, d\}, x_{1i} \geq x_{2i}, \\ \exists j \in \{1, 2, \dots, d\}, x_{1j} > x_{2j}. \end{aligned} \quad (7)$$

Otherwise, if either of the two conditions is not satisfied, then  $X_1$  does not dominate  $X_2$ .

*Definition 2 (skyline service).* For two manufacturing services  $MS_1$  and  $MS_2$ , if all the QoS attributes of  $MS_1$  are better than or equal to the corresponding ones of  $MS_2$ , and at least one attribute of  $MS_1$  performs strictly better than the corresponding attribute of  $MS_2$ , then  $MS_1$  dominates  $MS_2$ , which is denoted as  $MS_1 < MS_2$  [34]. That is to say,  $MS_1$  is better than  $MS_2$ . A skyline service refers to such a service that is not dominated by any other services.

The application of skyline operator can limit the size of the solution space of manufacturing service composition because the process of choosing skyline service eliminates those other services that could not possibly be selected as the final manufacturing service composition.

## 4. Improved Flower Pollination Algorithm

4.1. *Introduction to the Standard FPA.* FPA is a novel bionic evolutionary algorithm that was first proposed by Yang in 2012 [11]. The inspiration for FPA comes from the natural pollination process that takes place in flowering plants. In FPA, each flower stands for a feasible solution and the objective function value in Section 3.3 is regarded as the fitness value. To mimic the pollination process, there are two different pollination methods for each flower to choose: global or local pollination with switch probability  $p$ . The specific process is described as follows.

For each individual, there is a random number  $\text{rand}$ . If  $\text{rand} < p$ , then global pollination should be carried out. In the global pollination process, each flower updates its position according to the following equation [11]:

$$X_i^{t+1} = X_i^t + \gamma L(\lambda) (X_{\text{best}}^t - X_i^t), \quad (8)$$

where  $X_i^t$  and  $X_i^{t+1}$  are the old and new positions of  $i$ th flower, respectively,  $X_{\text{best}}^t$  is the best flower at current iteration  $t$ ,

which has the best fitness value in the whole population, and  $\gamma$  is the scaling factor that controls the step size of global pollination. Parameter  $L$ , the Lévy flight, is used as the strength of pollination in the basic FPA, and the step size  $L(\lambda)$  obeys the Lévy distribution:

$$L \sim \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}, \quad (s \gg s_0 > 0), \quad (9)$$

where  $\Gamma(\lambda)$  is the standard gamma function and  $\lambda$  is from 0.3 to 1.99 [35]. In the basic FPA, Yang [11] set the distribution factor  $\lambda$  as 1.5. This distribution is valid when large step  $s$  is bigger than 0.

If  $\text{rand} > p$ , the local pollination process should be carried out. Flower  $X_i^t$  obtains its new position  $X_i^{t+1}$  according to the difference between its old position and the position of two neighboring flowers  $X_p^t$  and  $X_q^t$ . This process is considered as local search, and the updating equation [11] is defined as

$$X_i^{t+1} = X_i^t + r (X_p^t - X_q^t), \quad (10)$$

where  $r$  is drawn from a uniform distribution  $[0, 1]$ , and it is considered as a local random walk.

After pollination is completed, the new individuals update their positions by comparing fitness values. If the fitness of  $X_i^{t+1}$  is better than that of  $X_i^t$ , the new position of  $i$ th flower will be replaced by  $X_i^{t+1}$ . Otherwise,  $i$ th flower remains at  $X_i^t$ .

### 4.2. Improvements of IFPA Compared with the Standard FPA.

Like many other evolutionary algorithms, basic FPA suffers from premature convergence and can easily fall into local optimum. In order to overcome these shortcomings, the authors propose an IFPA to extend the standard FPA in three aspects. Firstly, the whole population is divided into different groups on average, and an elite replacement strategy is introduced to replace the worst individual in each group. Secondly, the switch probability  $p$  and scaling parameter  $\gamma$  are adapted dynamically with the iterative process rather than remaining fixed like the basic FPA. This modification increases not only the global search ability but also the local search ability of basic FPA. Thirdly, in order to enhance its local search ability and obtain better optimal solution, the local search method is combined with the mutation and crossover operators of DE.

#### 4.2.1. Grouping Strategy and Elite Replacement Strategy.

In the original FPA, flowers search for the optimal solution in a single population only. This method may take a relatively long time and its global search ability is insufficient. Based on this idea and inspired by [32], the authors introduce a parallel grouping strategy to solve this problem effectively. Before starting the iteration, the whole population is divided into three different groups on average according to their fitness values. The detailed operation steps are described as follows. Firstly, the fitness is calculated and ranked. Secondly, the whole population is divided into three groups on average

according to their fitness. The first group consists of individuals with the best fitness, the third group consists of individuals with the worst fitness, and the remaining individuals belong to the second group. The iterative process is carried out in each group concurrently.

In addition, in order to improve the FPA convergence and the optimal solution quality, an elite replacement strategy is introduced. From each group, the best individual is selected as an elite individual. That is to say, after global and local pollination, the worst individual is replaced with the elite individual in each group. If there are duplicate individuals, then the duplicate one can be modified by selecting one dimension randomly to carry out mutation before starting the next iteration. In each group, FPA is carried out concurrently in the same way. Before executing the next generation, these groups are combined into a single population.

**4.2.2. Adaptive Switch Probability and Scaling Factor.** In standard FPA, Yang [11] set the switch probability  $p$  as 0.8. However, a fixed switch probability is insufficient. In order to retain a balance between global and local pollination in FPA, the switch probability is generated dynamically according to the fitness value of the current generation, which is represented by the objective function. The adaptive switch probability is obtained according to the following equation:

$$p = e^{(\max f - \min f) / \max f}, \quad (11)$$

where  $\max f$  and  $\min f$  are the maximum and minimum fitness values, respectively, of an individual in the current iteration.

For the same reason, in order to escape local optimum and avoid premature convergence in global pollination, the scaling factor  $\gamma$  is also generated dynamically according to the number of iterations. The calculation of scaling factor depends on the following formula:

$$\gamma = \frac{\max \text{iter} - t}{\max \text{iter}} \times 0.3 + 0.2, \quad (12)$$

where  $\max \text{iter}$  represents the maximum iterations and  $t$  is the current iterations. According to this formula, the value of scaling factor  $\gamma$  ranges from 0.2 to 0.5, and its value decreases with the increase of iterations.

**4.2.3. Modified Local Pollination Process by Combining with DE Algorithm.** In the local pollination process of  $X_i$ , the individuals  $X_p$  and  $X_q$  are selected randomly except for the rule that  $X_p$  and  $X_q$  should differ from the objective individual  $X_i$ . However, this method is not sufficiently farsighted and there is no direction of evolution to follow, which may lead the algorithm to fall into local convergence. In order to avoid this, the local search method is modified by combining it with DE. The detailed steps are as follows.

Firstly, the DE mutation operator [31] is applied in local search. For  $X_i^t$ , a new mutated individual  $V_i^t$  will be generated according to the following equation:

$$V_i^t = X_{\text{best}}^t + r(X_p^t - X_q^t), \quad (13)$$

where  $X_{\text{best}}^t$  is the best individual in current generation  $t$  in the whole population,  $X_p^t$  and  $X_q^t$  are different from  $X_i^t$  and belong to the same group with  $X_i^t$ , and  $r$  is a random number that is drawn from a uniform distribution  $[0, 1]$ .

Secondly, the crossover operator [31] is also introduced after mutation. The two-point crossover operator can be carried out as:

$$X_{i,j}^{t+1} = \begin{cases} V_{i,j}^t, & \text{if } l \leq j \leq h \\ X_{i,j}^t, & \text{else} \end{cases} \quad (14)$$

$$i = 1, 2, \dots, n; \quad j = 1, 2, \dots, d.$$

Here,  $X_{i,j}$  is  $j$ th dimension of  $i$ th individual  $X_i$ ;  $l$  and  $h$  are generated randomly from  $[1, d]$ . The mutated individual  $V_i$  obtained by mutation operator changes its location according to the above equation. This modification not only increases the local search ability but also improves the population diversity to avoid local optimal.

The pseudocode of improved FPA is summarized in Pseudocode 1.

**4.3. Representation Scheme of IFPA in Multiobjective Manufacturing Service Composition.** In order to solve the problem of multiobjective manufacturing service composition with IFPA, the representation scheme should be solved first. In this paper, population individuals are referred to as flowers. Each flower represents one possible solution of manufacturing service composition; it has  $N$  dimensions that stand for the number of subtasks. We use real encoding in this manuscript, with all numbers as integers. For example, a flower  $X = (3, 4, 2, 5, 1, 7)$  that contains six dimensions represents a manufacturing service composition solution that comprises six subtasks. The first dimension (value = 3) instructs the first subtask to select its third candidate service. The second dimension (value = 4) instructs the second subtask to select its fourth candidate service and so on for the remaining dimensions. In this paper, we use manufacturing service selection (MSS) to represent the candidate service in each service set, and  $\text{MSS}_{nm}$  stands for  $n$ th subtask selecting  $m$ th candidate service. Thus, the corresponding composition solution with this pollen can be expressed as  $(\text{MSS}_{13}, \text{MSS}_{24}, \text{MSS}_{32}, \text{MSS}_{45}, \text{MSS}_{51}, \text{MSS}_{67})$ .

**4.4. The Procedure of Using IFPA for Solving Multiobjective Manufacturing Service Composition Problems.** The whole procedure of the proposed algorithm is described as follows.

**Step 1 (selecting skyline services).** Firstly, specific user requirements are identified, including constraint conditions and preferences. Secondly, skyline services are selected according to the aforementioned definitions in Section 3.4 before IFPA is carried out.

**Step 2 (initializing).** Firstly, the parameters of the proposed method are set, including initial population size, maximum iterations, and number of groups. In addition,  $\lambda = 1.5$  is set in the Lévy flight. Secondly, an original population from the skyline service set is initialized.

```

(1) Initialize ( $t = 0$ )
Objective max  $f(x)$ 
Set parameters: population size  $N$ , maximum iterations max iter and  $\lambda$ 
Randomly generate an initial population of  $N$  flowers  $X(t) = (X_1^t, X_2^t, \dots, X_N^t)$ 
(2) Repeat ( $t = 1$  to max iter)
while ( $t < \text{max iter}$ )
Find the best individual  $X_{\text{best}}$  in current population
Divide the whole population into three groups
Define a switch probability  $p$  via  $p = e^{(\text{max } f - \text{min } f) / \text{max } f}$ 
  for  $i = 1 : N$ 
    if rand  $< p$ 
      Generate a step size  $L(\lambda)$ 
      Define a scaling factor  $\gamma$  via  $\gamma = ((\text{max iter} - t) / \text{max iter}) \times 0.3 + 0.2$ 
      Generate a step size  $L$  that obeys a Levy distribution
      Do global pollination via  $X_i^{t+1} = X_i^t + \gamma L(\lambda)(X_{\text{best}}^t - X_i^t)$ 
    else
      Do local pollination via  $X_i^{t+1} = X_i^t + r(X_p^t - X_q^t)$ 
      Execute crossover operation and find out  $X_i^{t+1}$ 
    end if
    Evaluate new individual
    If new individual is better, update its location
  end for
Carry out elite replacement strategy
Combine these three groups into one population
Update the best individual  $X_{\text{best}}$ 
end while
(3) Output
Output the best individual  $X_{\text{best}}$ 

```

PSEUDOCODE 1: Pseudocode of improved FPA.

*Step 3* (grouping strategy). Firstly, the fitness of each individual is calculated and ranked, and the population is divided into three groups according to Section 4.2.1. Then, the global optimal individual  $X_{\text{best}}$  and the elite individual in each group are selected. The iterative process is carried out in all groups concurrently.

*Step 4* (identifying the switch probability). For each generation, its switch probability  $p$  is calculated according to (11). A random number rand for each individual is generated and then compared with  $p$  to determine the pollination method.

*Step 5* (global pollination). If  $\text{rand} < p$ , then global pollination is chosen. The corresponding calculations are given in (8), (9), and (12).

*Step 6* (local pollination). If  $\text{rand} > p$ , then local pollination is chosen. The corresponding calculations are given in (13) and (14).

*Step 7* (updating locations). After completion of global and local pollination, the location of each individual is updated. If  $f(X_i^{t+1}) > f(X_i^t)$ , then  $i$ th individual updates its location; otherwise, it remains at old location.

*Step 8* (elite replacement strategy). After location updating is finished, the elite replacement strategy is employed to replace the worst individual in each group according to Section 4.2.1.

*Step 9* (merging). When all groups have finished the pollination process, the three groups are combined into a single population, and then the current best solution  $X_{\text{best}}$  is updated.

*Step 10* (stopping iteration). In general, there are two termination conditions for an evolutionary algorithm with an optimization problem: maximum iterations and fitness value. If either of the conditions is satisfied, then the iteration terminates and the best solution  $X_{\text{best}}$  is outputted; otherwise return to Step 3 and repeat until a termination condition is satisfied.

The flow chart of the proposed method is shown in Figure 1.

## 5. Experimental Simulation and Comparison

In order to verify the practicality and effectiveness of the IFPA, the proposed method is used to solve an illustrated problem in manufacturing service composition. A variety of comparative experiments with GA [4], DE [31], basic FPA [11], and extended FPA (EFPA) [32] were carried out to illustrate the superiority of the proposed method. All the experiments were carried out using a personal computer with Windows 7, an Intel Core 3-GHz CPU, and 2 GB of memory. All of the algorithms were programmed in C#.

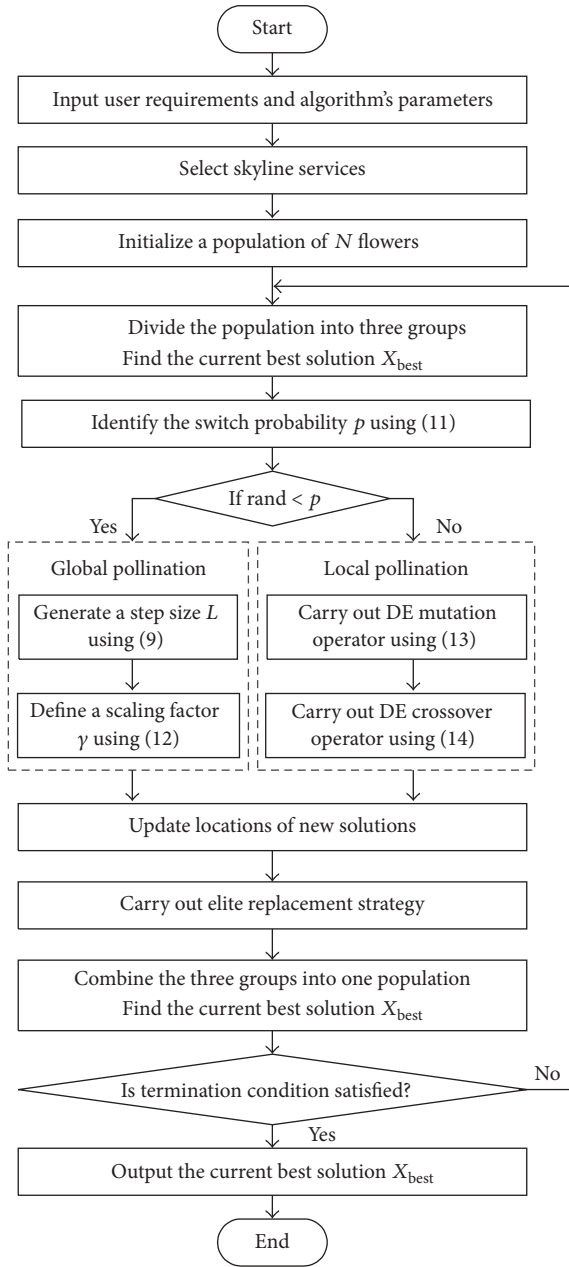


FIGURE 1: Flow chart of proposed method.

**5.1. Experimental Design.** A case study of manufacturing service composition with six subtasks was designed to illustrate the practicality and effectiveness of the proposed method. Although the example is relatively simple, it includes all the four structures of manufacturing service composition, as shown in Figure 2. The final fitness function of this model can be referred to Section 3.

The selection probabilities of subtasks 2 and 3 are 0.4 and 0.6, respectively. The cycle time of subtask 6 is five. The weight of each QoS attribute is set as 0.25, and the weights of QoS and Ec are 0.6 and 0.4, respectively. Each manufacturing service has four QoS attributes (time, cost, reliability, and availability) and one environmental index

(carbon emission). The maximum iterations are 50. There are two termination conditions: either it reaches the maximum number of iterations or it converges such that the successive difference in final average fitness is less than 0.001 for three consecutive generations of individuals [19].

**5.2. The Practicality of IFPA.** In order to verify the practicality of IFPA, the illustrative example shown in Section 5.3 is assigned with the practical manufacturing data. The partial candidate manufacturing service and corresponding attribute values are shown in Table 1. The final constraint conditions of the whole service composition are Total Time < 32 hrs, Total Cost < \$210, and Total Ec\* < 3. There are 15 candidate services in each manufacturing service set for each subtask. As for the proposed IFPA, the initial population size is 20. Figure 3 records the evolutionary tracks of the proposed method, including the best and average fitness values in each generation. Here, the horizontal axis represents iteration number and the vertical axis denotes the fitness of each generation. As shown in Figure 3, the IFPA stopped at the maximum iterations. And the optimal fitness value at the 50th iteration was 5.81; the best individual was  $X = (7, 1, 7, 1, 2, 5)$ . This means that the optimal service composition solution is  $(MSS_{17}, MSS_{21}, MSS_{37}, MSS_{41}, MSS_{52}, MSS_{65})$ . Thus, it can be concluded that the proposed approach can find the relative optimal solution and converges acceptably well.

In order to test the efficiency and superiority of the proposed method, IFPA is compared with other algorithms, including GA, DE, basic FPA, and EFPA, to solve the same problem described above. Figure 4 shows the different evolutionary tracks of these algorithms, and the orange dots represent the iterative termination points. In order to ensure the objectivity of the experiment, different weight combinations are taken into account, including not only the weight of cost, time, reliability, and availability but also the weight of QoS and environment index. From this figure, two results can be obtained. Firstly, the fitness values obtained from the proposed method are higher than those obtained by the other algorithms in most cases. In other words, the proposed method can obtain better solutions even in comparison with EFPA. Secondly, the proposed method converges faster than the other algorithms. It illustrates that the proposed algorithm has a strong search ability and fast convergence speed. Hence, it is an effective method to solve the problem of manufacturing service composition even under different weight combinations in the final objective function.

**5.3. The Effectiveness of IFPA.** In order to test the effectiveness and stability of the proposed method, the fitness values of the best manufacturing service composition solution in different algorithms are compared by changing the values of QoS properties and environmental index of candidate services, the number of candidate services in each service set, and the initial population size. In these experiments, the value ranges of QoS properties and environmental index are Cost = 0–50, Time = 0–10, Rel = 0.6–1, Ava = 0.4–1, and  $\theta = 0–5$ . The final constraint conditions of the whole service composition are the same as Section 5.2.



TABLE 1: The partial candidate manufacturing services and corresponding attribute values.

| Subtask              | Candidate manufacturing service | Cost (0, 50) | Time (0, 10) | Reliability (0.6, 1) | Availability (0.4, 1) | Carbon emission (0, 5) |
|----------------------|---------------------------------|--------------|--------------|----------------------|-----------------------|------------------------|
| Subtask <sub>1</sub> | MSS <sub>11</sub>               | 33.00        | 2.99         | 0.99                 | 0.48                  | 3.57                   |
|                      | MSS <sub>12</sub>               | 7.75         | 6.19         | 0.64                 | 0.42                  | 2.86                   |
|                      | MSS <sub>13</sub>               | 32.51        | 9.39         | 0.70                 | 0.95                  | 1.54                   |
| Subtask <sub>2</sub> | MSS <sub>21</sub>               | 29.04        | 4.18         | 0.60                 | 0.64                  | 0.87                   |
|                      | MSS <sub>22</sub>               | 3.79         | 7.38         | 0.65                 | 0.57                  | 3.69                   |
|                      | MSS <sub>23</sub>               | 28.54        | 0.58         | 0.70                 | 0.51                  | 1.48                   |
| Subtask <sub>3</sub> | MSS <sub>31</sub>               | 0.32         | 2.17         | 0.96                 | 0.86                  | 4.35                   |
|                      | MSS <sub>32</sub>               | 25.08        | 5.37         | 0.61                 | 0.79                  | 2.37                   |
|                      | MSS <sub>33</sub>               | 49.83        | 8.56         | 0.66                 | 0.73                  | 0.65                   |
| Subtask <sub>4</sub> | MSS <sub>41</sub>               | 21.61        | 0.15         | 0.92                 | 0.48                  | 2.95                   |
|                      | MSS <sub>42</sub>               | 46.36        | 3.35         | 0.97                 | 0.42                  | 0.29                   |
|                      | MSS <sub>43</sub>               | 21.11        | 6.55         | 0.62                 | 0.95                  | 4.45                   |
| Subtask <sub>5</sub> | MSS <sub>51</sub>               | 42.89        | 8.14         | 0.88                 | 0.70                  | 3.67                   |
|                      | MSS <sub>52</sub>               | 17.65        | 1.34         | 0.93                 | 0.64                  | 1.28                   |
|                      | MSS <sub>53</sub>               | 42.40        | 4.54         | 0.98                 | 0.57                  | 2.36                   |
| Subtask <sub>6</sub> | MSS <sub>61</sub>               | 39.43        | 2.93         | 0.79                 | 0.99                  | 4.32                   |
|                      | MSS <sub>62</sub>               | 38.93        | 9.33         | 0.89                 | 0.86                  | 0.57                   |
|                      | MSS <sub>63</sub>               | 13.68        | 2.53         | 0.94                 | 0.79                  | 3.75                   |

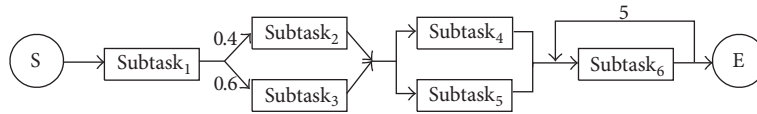


FIGURE 2: Case study of manufacturing service composition.

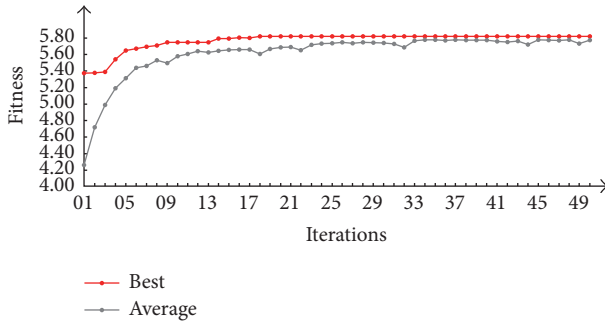


FIGURE 3: Evolutionary tracks of the proposed method.

The results shown in Figure 5 indicate that most often the proposed method obtains the best solutions with the highest fitness values under different number of candidate services in each service set. In the same way, Figure 6 shows that when the initial population size changes, the results obtained by the proposed method are better than those of the other methods in most cases. In order to ensure the objectivity of the experiment, different weight combinations are taken into account, including not only the weight of execution cost, execution time, reliability, and availability but also the weight of QoS and environment index.

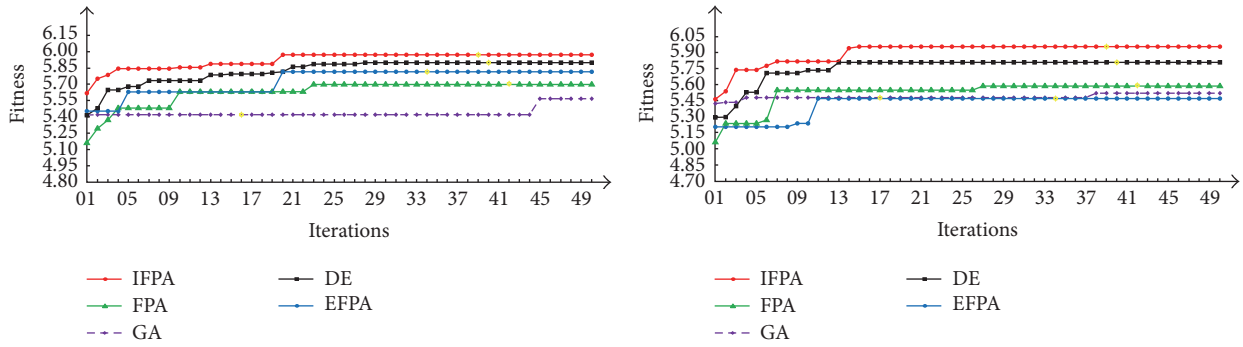
Based on all the experimental results, it is concluded that the proposed method searches better and converges

faster than the other algorithms evaluated in this paper, even under different weight combinations in the final objective function. The proposed method performs better in terms of both effectiveness and efficiency than the baseline algorithms of GA, DE, FPA, and EFPA in solving the problem of manufacturing service selection and composition.

## 6. Conclusions

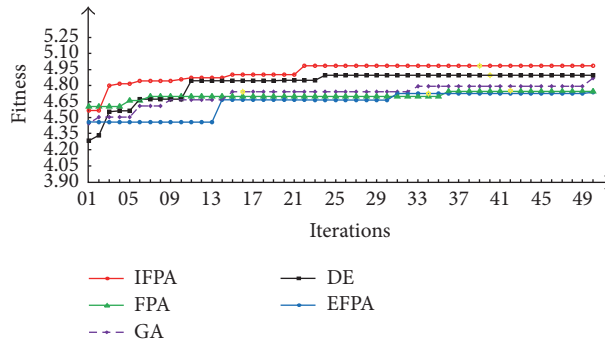
In this study, a new method called IFPA was proposed to solve the problem of manufacturing service selection and composition based on QoS and an environmental index. The main contributions of this paper are as follows:

- (i) The concept of skyline service is introduced in the manufacturing service composition model. The skyline operator can not only reduce the search space but also improve the quality of final optimal solutions.
- (ii) Three improvements were made to the basic FPA to produce the IFPA. First, a grouping strategy and elite replacement strategy were employed to improve the convergence speed of FPA. Second, two adaptive parameters were used to adjust the switch probability and control the step size of the global search. Third, the mutation and crossover operators of the DE algorithm were integrated with the local search of FPA to improve the performance of FPA.



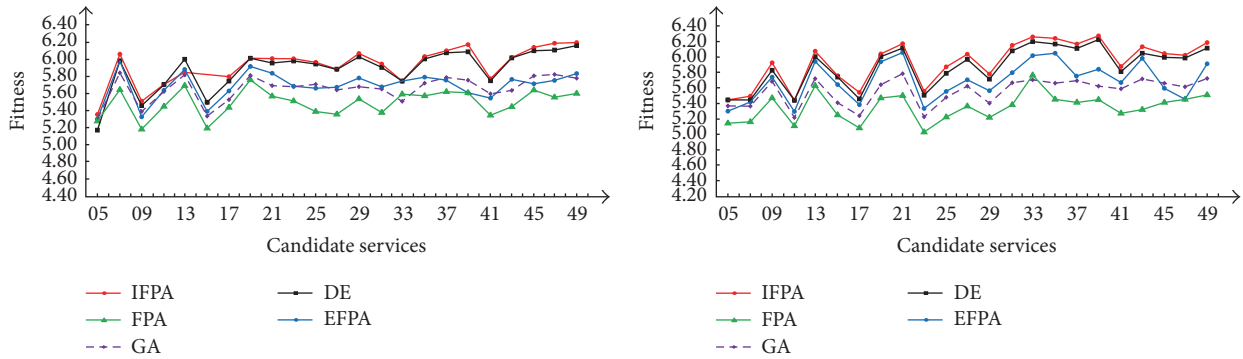
(a)  $w_1 = 0.25, w_2 = 0.25, w_3 = 0.25, w_4 = 0.25$ , and  $\lambda_1 = 0.6, \lambda_2 = 0.4$

(b)  $w_1 = 0.3, w_2 = 0.3, w_3 = 0.2, w_4 = 0.2$ , and  $\lambda_1 = 0.7, \lambda_2 = 0.3$



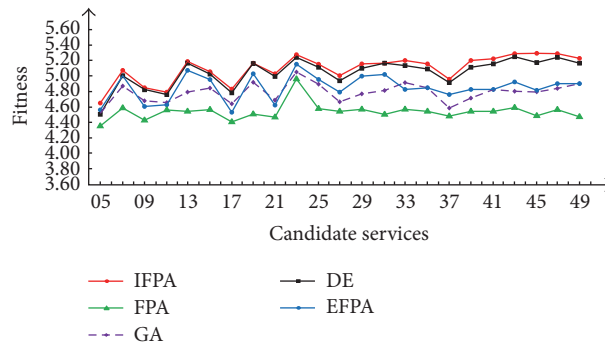
(c)  $w_1 = 0.4, w_2 = 0.1, w_3 = 0.3, w_4 = 0.2$ , and  $\lambda_1 = 0.8, \lambda_2 = 0.2$

FIGURE 4: Evolutionary tracks of different algorithms with different weight combinations.



(a)  $w_1 = 0.25, w_2 = 0.25, w_3 = 0.25, w_4 = 0.25$ , and  $\lambda_1 = 0.6, \lambda_2 = 0.4$

(b)  $w_1 = 0.3, w_2 = 0.3, w_3 = 0.2, w_4 = 0.2$ , and  $\lambda_1 = 0.7, \lambda_2 = 0.3$



(c)  $w_1 = 0.4, w_2 = 0.1, w_3 = 0.3, w_4 = 0.2$ , and  $\lambda_1 = 0.8, \lambda_2 = 0.2$

FIGURE 5: Performance of different algorithms under different number of candidate services with different weight combinations.

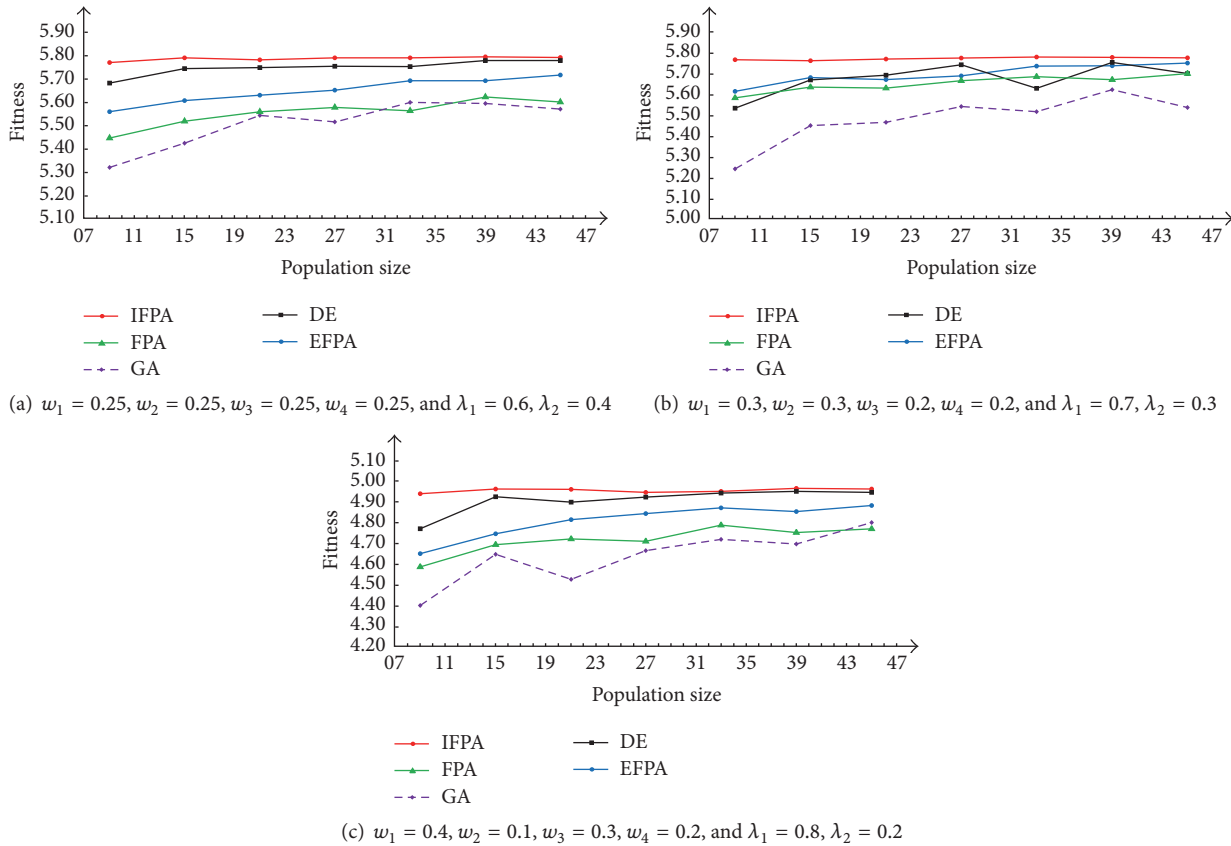


FIGURE 6: Performance of different algorithms under different initial population sizes with different weight combinations.

In this study, a multiobjective problem was transformed into a single objective problem. Nevertheless deficiencies exist that must be addressed. In a future study, the Pareto-optimal method can be considered to solve a real multiobjective problem instead of simply transforming it into a single objective problem. In addition, the authors will attempt to combine FPA with other evolutionary algorithms to improve the performance of basic FPA in order to solve the problem of manufacturing service composition.

**Competing Interests**

The authors declare that there are no competing interests regarding the publication of this paper.

**Acknowledgments**

The work has been supported by China National Natural Science Foundation (no. 51375429 and no. 51475410), Zhejiang Natural Science Foundation of China (no. LY17E050010), and Humanities and Social Sciences Research Project of Ministry of Education, China (no. 15YJC870008).

**References**

[1] F. Tao, D. Zhao, Y. Hu, and Z. Zhou, “Resource service composition and its optimal-selection based on particle swarm

optimization in manufacturing grid system,” *IEEE Transactions on Industrial Informatics*, vol. 4, no. 4, pp. 315–327, 2008.

[2] L. Zhang, Y. Luo, F. Tao et al., “Cloud manufacturing: a new manufacturing paradigm,” *Enterprise Information Systems*, vol. 8, no. 2, pp. 167–187, 2014.

[3] F. Tao, Y. Zuo, L. D. Xu, and L. Zhang, “IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1547–1557, 2014.

[4] Q. Li, R. Dou, F. Chen, and G. Nan, “A QoS-oriented Web service composition approach based on multi-population genetic algorithm for Internet of things,” *International Journal of Computational Intelligence Systems*, vol. 7, supplement 2, pp. 26–34, 2014.

[5] A. F. M. Huang, C.-W. Lan, and S. J. H. Yang, “An optimal QoS-based Web service selection scheme,” *Information Sciences*, vol. 179, no. 19, pp. 3309–3322, 2009.

[6] F. Tao, D. Zhao, H. Yefa, and Z. Zhou, “Correlation-aware resource service composition and optimal-selection in manufacturing grid,” *European Journal of Operational Research*, vol. 201, no. 1, pp. 129–143, 2010.

[7] J. Lartigau, X. Xu, L. Nie, and D. Zhan, “Cloud manufacturing service composition based on QoS with geo-perspective transportation using an improved Artificial Bee Colony optimisation algorithm,” *International Journal of Production Research*, vol. 53, no. 14, pp. 4380–4404, 2015.

[8] S. Borzsony, D. Kossmann, and K. Stocker, “The skyline operator,” in *Proceedings of the 17th International Conference on Data Engineering*, pp. 421–430, Heidelberg, Germany, April 2001.

- [9] S. Wang, Q. Sun, H. Zou, and F. Yang, "Particle Swarm Optimization with Skyline operator for fast cloud-based web service composition," *Mobile Networks and Applications*, vol. 18, no. 1, pp. 116–121, 2013.
- [10] D. Wang, Y. Yang, and Z. Mi, "A genetic-based approach to web service composition in geo-distributed cloud environment," *Computers & Electrical Engineering*, vol. 43, pp. 129–141, 2015.
- [11] X. Yang, "Flower pollination algorithm for global optimization," in *Proceedings of the 17th International Conference on Unconventional Computation and Natural Computation*, pp. 240–249, Orléans, France, September 2012.
- [12] Q. Z. Sheng, X. Qiao, A. V. Vasilakos, C. Szabo, S. Bourne, and X. Xu, "Web services composition: a decade's overview," *Information Sciences*, vol. 280, pp. 218–238, 2014.
- [13] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-aware middleware for Web services composition," *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 311–327, 2004.
- [14] F. Tao, Y. F. Hu, and Z. D. Zhou, "Study on manufacturing grid & its resource service optimal-selection system," *International Journal of Advanced Manufacturing Technology*, vol. 37, no. 9–10, pp. 1022–1041, 2008.
- [15] H. Zheng, Y. Feng, and J. Tan, "A fuzzy QoS-aware resource service selection considering design preference in cloud manufacturing system," *International Journal of Advanced Manufacturing Technology*, vol. 84, no. 1–4, pp. 371–379, 2016.
- [16] Y. Hao, Y. Zhang, and J. Cao, "A novel QoS model and computation framework in web service selection," *World Wide Web*, vol. 15, no. 5–6, pp. 663–684, 2012.
- [17] F. Tao, L. Zhang, Y. Liu, Y. Cheng, L. Wang, and X. Xu, "Manufacturing service management in cloud manufacturing: overview and future research directions," *Journal of Manufacturing Science and Engineering*, vol. 137, no. 4, pp. 1–11, 2015.
- [18] M. Alrifai, D. Skoutas, and T. Risse, "Selecting skyline services for QoS-based web service composition," in *Proceedings of the 19th International World Wide Web Conference (WWW '10)*, pp. 11–20, North Carolina, USA, April 2010.
- [19] W. Y. Zhang, S. Zhang, M. Cai, and J. X. Huang, "A new manufacturing resource allocation method for supply chain optimization using extended genetic algorithm," *International Journal of Advanced Manufacturing Technology*, vol. 53, no. 9–12, pp. 1247–1260, 2011.
- [20] F.-C. Pop, D. Pallez, M. Cremene, A. G. B. Tettamanzi, M. Suci, and M. Vaida, "QoS-based service optimization using differential evolution," in *Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference (GECCO '11)*, pp. 1891–1898, Ireland, July 2011.
- [21] X. Wei and H. Liu, "A cloud manufacturing resource allocation model based on ant colony optimization algorithm," *International Journal of Grid Distribution Computing*, vol. 8, no. 1, pp. 55–66, 2015.
- [22] Z.-J. Wang, Z.-Z. Liu, X.-F. Zhou, and Y.-S. Lou, "An approach for composite web service selection based on DGQoS," *International Journal of Advanced Manufacturing Technology*, vol. 56, no. 9–12, pp. 1167–1179, 2011.
- [23] A. Y. Abdelaziz and E. S. Ali, "Static VAR compensator damping controller design based on flower pollination algorithm for a multi-machine power system," *Electric Power Components and Systems*, vol. 43, no. 11, pp. 1268–1277, 2015.
- [24] A. Abdelaziz, E. Ali, and S. A. Elazim, "Implementation of flower pollination algorithm for solving economic load dispatch and combined economic emission dispatch problems in power systems," *Energy*, vol. 101, pp. 506–518, 2016.
- [25] G. Bekdaş, S. M. Nigdeli, and X.-S. Yang, "Sizing optimization of truss structures using flower pollination algorithm," *Applied Soft Computing Journal*, vol. 37, pp. 322–331, 2015.
- [26] I. El-henawy and M. Ismail, "An improved chaotic flower pollination algorithm for solving large integer programming problems," *Journal of Digital Content Technology and Its Applications*, vol. 8, no. 3, pp. 72–81, 2014.
- [27] R. Wang, Y. Zhou, S. Qiao, and K. Huang, "Flower pollination algorithm with bee pollinator for cluster analysis," *Information Processing Letters*, vol. 116, no. 1, pp. 1–14, 2016.
- [28] S. Ouadfel and A. Taleb-Ahmed, "Social spiders optimization and flower pollination algorithm for multilevel image thresholding: a performance study," *Expert Systems with Applications*, vol. 55, pp. 566–584, 2016.
- [29] Z. A. Dahi, C. Mezioud, and A. Draa, "On the efficiency of the binary flower pollination algorithm: application on the antenna positioning problem," *Applied Soft Computing*, vol. 47, pp. 395–414, 2016.
- [30] A. Draa, "On the performances of the flower pollination algorithm—qualitative and quantitative analyses," *Applied Soft Computing*, vol. 34, pp. 349–371, 2015.
- [31] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [32] R. Wang, Y. Zhou, C. Zhao et al., "A hybrid flower pollination algorithm based modified randomized location for multi-threshold medical image segmentation," *Bio-Medical Materials and Engineering*, vol. 26, supplement 1, pp. S1345–S1351, 2015.
- [33] S. X. Sun and J. Zhao, "A decomposition-based approach for service composition with global QoS guarantees," *Information Sciences*, vol. 199, pp. 138–153, 2012.
- [34] M. Hoseini, H. Hosseinpour, and B. Bastaaee, "A new multi objective optimization approach in distribution systems," *Optimization Letters*, vol. 8, no. 1, pp. 181–199, 2014.
- [35] H. M. Dubey, M. Pandit, and B. K. Panigrahi, "A biologically inspired modified flower pollination algorithm for solving economic dispatch problems in modern power systems," *Cognitive Computation*, vol. 7, no. 5, pp. 594–608, 2015.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

