DE GRUYTER OPEN

Int. J. Appl. Math. Comput. Sci., 2017, Vol. 27, No. 2, 417–433
DOI: 10.1515/amcs-2017-0029

amcs

STOCHASTIC FRACTAL BASED MULTIOBJECTIVE FRUIT FLY OPTIMIZATION

CILI ZUO [a], LIANGHONG WU [a,*], ZHAO-FU ZENG [a], HUA-LIANG WEI [b]

[a] School of Information and Electrical Engineering
Hunan University of Science and Technology, Xiangtan, Hunan 411201, PR China
e-mail: lhwu@hnust.edu.cn

[b] Department of Automatic Control and Systems Engineering
University of Sheffield, Sheffield S1 3JD, UK

The fruit fly optimization algorithm (FOA) is a global optimization algorithm inspired by the foraging behavior of a fruit fly swarm. In this study, a novel stochastic fractal model based fruit fly optimization algorithm is proposed for multiobjective optimization. A food source generating method based on a stochastic fractal with an adaptive parameter updating strategy is introduced to improve the convergence performance of the fruit fly optimization algorithm. To deal with multiobjective optimization problems, the Pareto domination concept is integrated into the selection process of fruit fly optimization and a novel multiobjective fruit fly optimization algorithm is then developed. Similarly to most of other multiobjective evolutionary algorithms (MOEAs), an external elitist archive is utilized to preserve the nondominated solutions found so far during the evolution, and a normalized nearest neighbor distance based density estimation strategy is adopted to keep the diversity of the external elitist archive. Eighteen benchmarks are used to test the performance of the stochastic fractal based multiobjective fruit fly optimization algorithm (SFMOFOA). Numerical results show that the SFMOFOA is able to well converge to the Pareto fronts of the test benchmarks with good distributions. Compared with four state-of-the-art methods, namely, the non-dominated sorting generic algorithm (NSGA-II), the strength Pareto evolutionary algorithm (SPEA2), multi-objective particle swarm optimization (MOPSO), and multiobjective self-adaptive differential evolution (MOSADE), the proposed SFMOFOA has better or competitive multiobjective optimization performance.

Keywords: multiobjective optimization, fruit fly optimization algorithm, stochastic fractal.

1. Introduction

Multiobjective optimization problems (MOPs) widely exist in all areas of science, engineering, economics, finance and technology, where the optimal decisions need to be made in the presence of trade-offs between two or even more conflicting objectives. In comparison with single-objective optimization problems, MOPs are more difficult to solve as the conflicting objectives must be optimized simultaneously. Instead of finding a single optimal solution, the purpose of MOPs is to find a set of optimal solutions, largely known as the Pareto optimal set, that are uniformly distributed along the whole Pareto front. Since most classical optimization algorithms can only obtain a Pareto optimal solution in one run, to acquire a Pareto optimal set, these optimization algorithms have to be run many times. Consequently, the efficiency of these optimization algorithms is very low for solving MOPs.

Evolutionary algorithms (EAs) are ideal and already popular for solving MOPs since the population-based algorithms can obtain a set of potential solutions in a single run and are robust to the shape of the Pareto front and the underlying objective function characteristics. To obtain the Pareto optimal solutions, various MOEAs have been proposed over the past few decades (Zhou *et al.*, 2011). Generally, MOEAs can be categorized into three groups, namely, dominance-based methods, scalarization-based methods, and performance indicator-based approaches (Cheng *et al.*, 2015; Denysiuk *et al.*, 2015). Among these, dominance-based approaches such as SPEA2 (Zitzler *et al.*, 2001), NSGA-II (Deb *et al.*, 2002; Ben Aicha *et al.*, 2013), NSGA-III (Deb and Jain, 2014), and NSLS (Chen *et al.*, 2015) have been

[*] Corresponding author

probably the most commonly used approaches, which calculate an individual's fitness on the basis of the Pareto dominance relation.

Scalarization-based or weighted aggregation-based approaches (Ishibuchi *et al.*, 2006; Zhang and Li, 2007; Liu *et al.*, 2014; Wang *et al.*, 2015; Mei *et al.*, 2011) have been increasingly popular in recent years owing to their computational efficiency. These methods use traditional mathematical techniques to aggregate multiple objectives into a single parameterized objective to assign scalar fitness values to population members. The most representative scalarization-based method is MOEA/D (Ishibuchi *et al.*, 2006), which has received increasing attention due to its computational simplicity and attractive search performance (Denysiuk *et al.*, 2015; Wang *et al.*, 2015). Meanwhile, indicator-based approaches (Zitzler and Künzli, 2004; Bader and Zitzler, 2011; Rodríguez Villalobos and Coello Coello, 2012; Menchaca-Mendez and Coello, 2015), as a relatively recent trend, employ performance indicators for fitness assignment. More studies about the development and application of MOEAs are well reviewed by Zhou *et al.* (2011) or Eiben and Smith (2015).

Swarm intelligence, which is inspired by nature, especially biological systems, has become a hot topic in artificial intelligence in the past decades (Bonabeau *et al.*, 1999; Bilski and Wojciechowski, 2014). More and more researchers are interested in this new exciting way of achieving a form of swarm intelligence, i.e., the emergent collective intelligence of groups of simple agents, and various swarm intelligence based optimization algorithms are proposed such as particle swarm optimization (PSO) (Kennedy, 2011), artificial bee colony optimization (ABC) (Karaboga and Akay, 2009), ant colony optimization (ACO) (Dorigo *et al.*, 1996), the firefly algorithm (Yang, 2010), cuckoo search (Rajabioun, 2011), the grey wolf optimizer (Mirjalili *et al.*, 2014), etc. Similarly to a social insect or an animal metaphor in their problem solving, swarm intelligence based methods have been widely applied to many aspects in real-world problem solving and have produced very promising results. Accordingly, many multiobjective optimization algorithms have been developed and proposed (Mirjalili *et al.*, 2014).

The fruit fly optimization algorithm (FOA), proposed by Pan (2011), is a global optimization algorithm inspired by the foraging behavior of the fruit fly. Compared with other swarm intelligence based algorithms, the FOA has advantages of being easy to understand, involving fewer control parameters, and the simplicity of computational process (Pan, 2011; 2012). As a novel optimization algorithm, the FOA has gained much attention and has been successfully applied in many single objective optimization problems in recent years, including annual power load forecasting (Pan, 2012), analysis of the service satisfaction in web auction logistics service (Li *et al.*, 2013), PID controller tuning (Lin, 2013; Li *et al.*, 2012), neural network training (Sheng and Bao, 2013), the multidimensional knapsack problem (Chen *et al.*, 2013), the joint replenishment problem (Wang *et al.*, 2013), medical image classification (Wang *et al.*, 2015), the homogeneous fuzzy series-parallel redundancy allocation problem (Mousavi *et al.*, 2016), gasification process optimization (Shen *et al.*, 2016), and parameter identification (Niu *et al.*, 2015). Moreover, to enhance the convergence performance, a number of improved FOAs have also been proposed (Yuan *et al.*, 2014; 2015, Shan *et al.*, 2013; Pan *et al.*, 2014; Mitić *et al.*, 2015; Wu *et al.*, 2015). However, so far not much work on MOPs has been reported in the literature.

In this paper, an improved fruit fly optimization algorithm based on a stochastic fractal for multi-objective optimization, which is referred to as the SFMOFOA, is proposed. To improve the search efficiency of the FOA, a new food source generating method based on a stochastic fractal is proposed, and an adaptive parameter updating strategy is introduced to dynamically adjust the search range. To solve multiobjective optimization problems, the Pareto domination concept is integrated with the FOA's selection process and a novel multiobjective FOA is then developed. Similarly to most of existing MOEAs, an external elitist archive is utilized in the proposed SFMOFOA to preserve the nondominated solutions found so far during the evolution process. To let the elitists in the archive to be uniformly distributed along the whole Pareto front, a normalized nearest neighbor distance based density estimation strategy is adopted. Eighteen benchmark MOPs are used to evaluate the effectiveness of the proposed method. Compared with four state-of-the-art methods, namely, NSGA-II (Deb *et al.*, 2002), SPEA2 (Zitzler *et al.*, 2001), MOPSO (Coello *et al.*, 2004), and MOSADE (Wang *et al.*, 2010), the numerical results indicate that the proposed SFMOFOA is able to obtain a better or competitive convergence performance.

The rest of this paper is organized as follows. In Section 2, some basic concepts of MOPs are briefly proposed. The basic FOA is introduced in Section 3. Thereafter, the detailed descriptions of the framework of the SFMOFOA algorithm are proposed in Section 4. The experimental design and numerical comparisons are illustrated in Section 5. Finally, Section 6 gives the concluding remarks.

2. Basic concepts of MOPs

With no loss of generality, consider the general multiobjective optimization problems which are mathematically described as follows:

Minimize

$$\begin{aligned}&F(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})], \quad \mathbf{x} \in \mathbb{R}^n,\\ &\text{subject to}\\ &g_i(\mathbf{x}) \le 0, \quad i = 1, 2, \dots, m,\\ &h_i(\mathbf{x}) = 0, \quad j = 1, 2, \dots, p,\end{aligned} \tag{1}$$

where $F(\mathbf{x})$ is the objective vector to be optimized, and k is the number of objective functions, $g_i(\mathbf{x})\,(i = 1, 2, \dots, m)$ are the set of inequality constraints, $h_j(\mathbf{x})\,(j = 1, 2, \dots, p)$ are the set of equality constraints. The elements of the vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$ are the decision variables, and the symbol $\mathbb{R}^n$ represents the n-dimensional Euclidean space. The multiobjective optimization problem is to determine a particular set of solutions, which yield the optimum values for all the objective functions satisfying all the inequality and equality constraints.

To better understand multiobjective optimization, the basic Pareto concepts of dominance, the Pareto optimal set (PS) and the Pareto front (PF) are defined as follows (Van Veldhuizen and Lamont, 1998).

Definition 1. (*Concept of dominance*) A vector $\mathbf{u} = (u_1, u_2, \dots, u_k)$ is said to dominate $\mathbf{v} = (v_1, v_2, \dots, v_k)$ (which is denoted by $\mathbf{u} \prec \mathbf{v}$) if $\forall i \in \{1, 2, \dots, k\}, u_i \le v_i \wedge \exists i \in \{1, 2, \dots, k\} : u_i < v_i$.

Definition 2. (*Pareto optimal solution*) A point $\mathbf{x}^* \in \Omega$ is Pareto optimal if no other $\mathbf{x} \in \Omega$ satisfies $F(\mathbf{x}) \prec F(\mathbf{x}^*)$.

Definition 3. (*Pareto optimal set*) For a given MOP $F(\mathbf{x})$, the Pareto optimal set P^* is defined as $P^* = \{\mathbf{x} \in \Omega | \nexists \mathbf{x}' \in \Omega : F(\mathbf{x}') \prec F(\mathbf{x})\}$.

Definition 4. (*Pareto front*) For a given MOP $F(\mathbf{x})$ and a Pareto optimal set P^*, the Pareto front PF^* is defined in the objective space as $PF^* = \{F(\mathbf{x}) | \mathbf{x} \in P^*\}$.

Most MOPs may have many or even infinitely many Pareto optimal solutions. Generally, the main goals of MOEAs are: (i) to find a set of approximate solutions that are as close as possible to the true Pareto front, and (ii) to keep the obtained solutions to be spread along the whole Pareto front as uniformly as possible. In this paper, a novel normal cloud model based multiobjective fruit fly optimization algorithm is proposed to solve the MOPs.

3. Basic fruit fly optimization algorithm

The FOA is a method for searching global optimum based on the food foraging behavior of fruit flies (Pan, 2011), which live in the temperate and tropical climate zones, and which are superior to other species in osphresis and vision. When a fruit fly decides to go for hunting, it will fly randomly to find the location guided by a particular odor. While in searching, a fruit fly also sends and receives information from its neighbors and makes a comparison with the current best location and fitness (Yuan *et al.*, 2015). The food finding process of the fruit fly is as follows: firstly, it smells the food source by using olfactory sensors, and flies towards that location; secondly, after it gets close to the food location, the sensitive vision is also used for finding food and other fruit flies' flocking location, and then it flies in that direction. According to the food finding procedure of the fruit fly swarm, the FOA can be divided into three parts, namely, the initialization of parameters and the population location, the osphresis-based search and the vision-based search phase.

3.1. Initialization.

The main parameters of the FOA are the maximum iteration number T, the population size NP, and the range of the random flight distance R_f. The fruit fly swarm location $(X_{\text{axis}}, Y_{\text{axis}})$ is randomly initialized in the search space as:

$$X_{\text{axis}} = \text{rand} \times (\text{UB-LB}) + \text{LB}, \tag{2}$$

$$Y_{\text{axis}} = \text{rand} \times (\text{UB-LB}) + \text{LB}, \tag{3}$$

where 'rand' is a random function which returns a value from the uniform distribution on the interval $[0, 1]$, UB and LB are the upper and lower bounds of the fruit fly swarm location in the two-dimensional searching space, respectively.

3.2. Osphresis-based search.

In the osphresis-based searching phase, a population of NP food sources is generated randomly around the current fruit fly swarm location. First, a random flight direction is chosen and the distance for an individual fruit fly to find food is given by

$$X_i = X_{\text{axis}} + \xi \times R_f, \tag{4}$$

$$Y_i = Y_{\text{axis}} + \xi \times R_f, \tag{5}$$

where $i = 1, 2, \dots, \text{NP}$, ξ is a random function which returns a value in the range of $[-1, 1]$.

Then, for each individual calculate the distance of the food location to the origin (Dist_i), and further calculate the smell concentration judgment value (S_i). The value of S_i is the reciprocal of the distance of food location to the origin, that is,

$$\text{Dist}_i = \sqrt{X_i^2 + Y_i^2}, \tag{6}$$

$$S_i = \frac{1}{\text{Dist}_i} \tag{7}$$

Finally, calculate the smell concentration (Smell_i) of the individual fruit fly location by substituting the smell concentration judgment value (S_i) into the smell concentration judgment function (or called fitness function):

$$\text{Smell}_i = \text{Fitness}(S_i) \tag{8}$$

3.3. Vision-based search. In the vision-based searching phase, the FOA carries out a greedy selection procedure. Firstly, find an individual with the maximum smell concentration (the maximum value of Smell) among the fruit fly swarm. This can be represented as:

$$[\text{bestSmell,bestIndex}] = \min(\text{Smell}). \tag{9}$$

Then, compare the current maximum smell concentration value (bestSmell) with the value in history (smellBest). If bestSmell $<$ smellBest, update smellBest with bestSmell and the fruit fly swarm flies towards that location with the maximum smell concentration value by using vision as below:

$$\text{smellBest} = \text{bestSmell}, \tag{10}$$

$$X_{\text{axis}} = X(\text{bestIndex}), \tag{11}$$

$$Y_{\text{axis}} = Y(\text{bestIndex}). \tag{12}$$

The osphresis-based searching phase and the vision-based searching phase are repeated, until the smell concentration has no improvement any more, or the number of iteration reaches the maximum number.

4. Stochastic fractal based multiobjective FOA

The core idea that distinguishes the FOA from other swarm intelligence based optimization algorithms is the expression of smell concentration judgment value. However, it can be observed from (6) and (7) that the value of Dist_i varies in a large range while the corresponding value of S_i is very small. Therefore, the original FOA can only solve problems that have optimal solutions in a small vicinity of the origin (Niu *et al.*, 2015). The reason why the original FOA is only able to find solutions close to the origin and cannot solve complex optimization problems (Wu *et al.*, 2015b). To overcome this shortcoming of the FOA, a new food source generating method based on stochastic fractals is proposed in this section. In the following, the stochastic fractal based the FOA is developed and adopted to the multiobjective optimization problems by combining the concept of Pareto dominance with the external elitist archive strategy.

4.1. Random fractals. The concept of a fractal was first proposed by Mandelbrot (1983). A fractal is a natural phenomenon or a mathematical set that exhibits a repeating pattern that displays at every scale. It need not exhibit exactly the same structure at all scales, but the same type of structures must appear on all scales (Vicsek and Gould, 1989). Some stochastic method can provide a better model for naturally occurring fractals than sets obtained in a nonrandom way (Voss, 1991). For example, random fractals can be diffused by modifying the iteration process via stochastic rules (Vicsek and Gould, 1989; Witten and Sander, 1983). The characteristic of selfsimilarity and randomness displayed by the procedure of diffusion of random fractals is suitable for the optimizer and it has been applied to some optimization algorithms, such as the stochastic fractal search (SFS) algorithm (Salimi, 2015), the penalty guided stochastic fractal search (PSFS) approach (Mellal and Zio, 2016) and the decremental differential stochastic fractal evolutionary algorithm (dDSF-EA) (Awad *et al.*, 2016). Some random fractals, such as the clusters describing a bacterial colony, can be generated by a physically motivated model called diffusion limited aggregation (DLA) (Witten and Sander, 1983). To simulate the diffusion process, a mathematical algorithm like the random walk can be employed. In the fractal search, the Levy flight and the Gaussian walk are commonly used to model the DLA growth (Salimi, 2015). The Gaussian walk based stochastic fractal can be described as follows:

$$\begin{aligned} X_i^q = {} & \text{Gaussian}(X_i, |\beta \times \text{BP}|) \\ & + (\gamma \times \text{BP} - \gamma' \times X_i), \end{aligned} \tag{13}$$

$$\beta = \frac{\log(g)}{g}, \tag{14}$$

where q is the number of particles acquired from the diffusion of the main particle, g is the number of iterations, $\text{Gaussian}(x_i, |\text{BP}|)$ is the Gaussian distribution, BP is mentioned as the position of the best point, γ and γ' are random parameters between 0 and 1. The particle diffusion with random different positions is shown in Fig. 1.

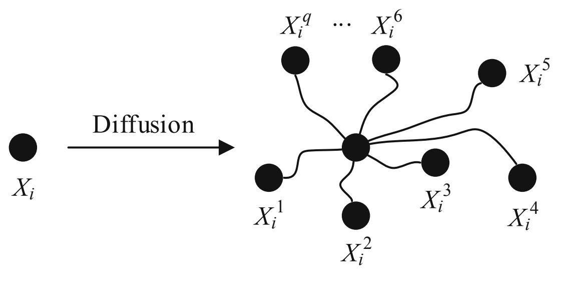

Fig. 1. Process of diffusing a particle.

4.2. Position update based on a stochastic fractal. From Eqns. (6) and (7), it can be observed that the basic fruit fly optimization algorithm uses the uniformly distributed random location method to update the position of all fruit flies. This method has its advantages of simple calculation, but it is not an effective search method. The random fractal displays a certain degree of selfsimilarity but need not exhibit exactly the same structure at all scales (Yamaguchi *et al.*, 1997). That is to say, it has both tendentiousness and randomness which are more beneficial for improving the search efficiency. Therefore, a new position update mechanism based on the diffusion process of stochastic fractal is proposed to enhance the search efficiency of the FOA.

According to the investigation by Salimi (2015), the Gaussian walk is more promising to obtain a better global final result. Therefore, here the Gaussian walk is used to update the fruit fly's positions as shown in (14),

$$X_{ij}^{'} = \begin{cases} \text{Gaussian}(X_{bj}, \delta) + (\gamma \times X_{r1j} \\ -\gamma' \times X_{r2j}), & \text{if } j = d, \\ X_{ij}, & \text{otherwise.} \end{cases} \tag{15}$$

where $i = 1, 2, \ldots, \text{NP}, j = 1, 2, \ldots, n$, X_b is the best position of the current population, X_{r1} and X_{r2} are randomly selected individuals from the fruit fly population, d is a random integer in the range of $[1, n]$, the standard deviation δ is used to represent the search range.

In order to guarantee a global search, a large search range is desirable and may be necessary in the early search stage. With the evolution of the swarm, especially in a late search stage, a small search radius is appropriate for the local exploitation and solution refining. Therefore, an adaptive strategy of δ that dynamically varies with iterations is adopted to balance the global exploration and local exploitation as below:

$$\delta = \left| \frac{\log(t+1)}{t^{\alpha}} \times (X_i - X_b) \right|, \tag{16}$$

where t is the iteration, $\alpha > 0$ is a positive integer and $\alpha = 2$ is used here.

Moreover, similarly to Yuan *et al.* (2014) or Pan *et al.* (2014), the smell concentration judgment value is not used and the fitness is evaluated directly in the decision space to avoid the limitations of the original definition of smell concentration judgement.

4.3. Pareto dominance selection mechanism.

A fundamental problem in multiobjective optimization is to compare candidate solutions. To adopt and develop the basic FOA to multiobjective optimization, it needs to modify the dominance selection mechanism since there are two or more conflicting objective functions to be evaluated and compared simultaneously. Generally, there are three approaches to compare two of conflicting multiobjective solutions, i.e., dominance-based methods, scalarization-based methods, and performance indicator-based methods. Here, the selection operation is based on the concept of Pareto dominance, the most popular selection mechanism in MOEAs.

For the fruit fly i in the population, once a new food source (solution $X_i^{'}$) is generated, there will be a choice between the current solution X_i and the new solution $X_i^{'}$. According to the Pareto dominance relation, there are three possibilities:

(i) X_i dominates $X_i^{'}$,
(ii) X_i is dominated by $X_i^{'}$,
(iii) X_i and $X_i^{'}$ are nondominated with each other.

Based on the above three possibilities, the dominance selection mechanism can be defined as

$$X_i(t+1) = \begin{cases} X_i(t) & \text{if } X_i(t) \prec X_i^{'}(t), \\ X_i^{'}(t) & \text{if } X_i^{'}(t) \prec X_i(t), \\ LC(X_i(t), X_i^{'}(t)) & \text{if } X_i(t) \not\prec X_i^{'}(t) \\ & \quad \wedge X_i^{'}(t) \not\prec X_i(t). \end{cases} \tag{17}$$

where $LC(X_i(t), X_i^{'}(t))$ indicates the less crowded one between $X_i(t)$ and $X_i^{'}(t)$ with respect to the external elitist archive. The crowding degree estimation method will be discussed in the next subsection.

4.4. Update the external archive.

Many MOEAs (Zitzler *et al.*, 2001; Deb *et al.*, 2002; Deb and Jain, 2014) have shown that it is necessary to apply an external archive to retain nondominated solutions because a nondominated solution in the current iteration is not always a nondominated solution in later evolutions. In the proposed algorithm, an external elitist archive is therefore used to keep the nondominated solutions found so far during evolution. Initially, the external elitist archive is empty, and it will be updated when new nondominated solutions are generated at each of iterations. There are three cases when comparing a newly generated nondominated solution with the individuals in the current archive, namely,

(i) If the new solution is dominated by any individual in the external archive, the new solution is rejected.

(ii) If the new solution dominates some individuals of the archive, then the dominated individuals are replaced by the new solution.

(iii) If the new solution is mutually non-dominated with the individuals in the archive, then it joins in the archive.

The pseudocode of external elitist archive updating is shown in Algorithm 1.

When the population of the external archive exceeds its maximum capacity, the diversity maintenance strategy (Zitzler *et al.*, 2001; Deb *et al.*, 2002) is used to keep the external archive at its maximum size. There are two representative diversity maintenance strategies in the EMO community, i.e., the crowding distance-based strategy in NSGA-II and the k-nearest neighbor distance-based strategy in SPEA2. The

Algorithm 1. External elitist archive updating.

```
1: if $X_i^{'}(t)$ dominate $X_i(t)$ or $X_i^{'}(t)$ and $X_i(t)$ are nondominated then
2:     if $X_i^{'}(t)$ is dominated by an individual(s) of the external archive then
3:         Reject $X_i^{'}(t)$ enter the archive
4:     else if $X_i^{'}(t)$ dominates some individual(s) of the archive then
5:         Remove the dominated individuals and $X_i^{'}(t)$ enters the archive
6:     else
7:         $X_i^{'}(t)$ joins in the archive
8:     end if
9: end if
```

crowding distance is an ideal measure for bi-objective optimization problem, but it is not so effective on optimization problems with three or more objectives. On the contrary, the nearest neighbor distance estimation strategy is a very effective diversity measure for problems with three and many objectives. However, the original nearest neighbor distance cannot accurately reflect the crowding degree of a problem whose Pareto front is composed of multiple parts and the covered ranges of different parts are significantly different in one or more objective(s). To overcome the above mentioned disadvantage, in this paper, the values of each objective function are normalized before calculating the distance from each solution to elements in the external archive,

$$f'_{ij} = \frac{f_{ij}}{f_j^{\max} - f_j^{\min}}. \tag{18}$$

where $i = 1, 2, \ldots, N, j = 1, 2, \ldots, k$, f_{ij} is the original objective function value, f'_{ij} is the normalized objective function value, $f_j^{\max}$ and $f_j^{\min}$ are the maximum and minimum objective function values of objective j, N is the number of external archive individuals, and k is the number of objective functions.

4.5. Constraint handling. For optimization problems with constraints, several methods have been proposed in the last few decades (Michalewicz and Schoenauer, 1996; Fonseca and Fleming, 1998; Asafuddoula *et al.*, 2015). The penalty-function approach is one of the most frequently used approaches for constraint handling. However, the choice of an appropriate penalty factor(s) is problem dependent and nontrivial. To overcome the drawbacks of penalty-function method, some researchers have suggested the use of multiobjective optimization to handle constraints in EAs (Fonseca and Fleming, 1998; Aguirre *et al.*, 2004). The feasibility-first scheme is another type of the constraint handling approach where a feasible solution is always preferred over an infeasible solution. The common forms of preference rules are the following: (i) any feasible solution is preferred over an infeasible solution, (ii) among two feasible solutions, the one with a better objective is preferred, and (iii) among two infeasible solutions, the one with a lowest constraint violation is preferred (Deb *et al.*, 2002; Deb, 2000). Moreover, Fletcher's filter has been recently used as a general methodology to handle constraints in EAs (Rafajłowicz and Rafajłowicz, 2012; Rafajłowicz, 2013).

In this paper, a concept of sequence constrained domination is used to handle constrained MOPs by means of an improved feasibility-first scheme (Asafuddoula *et al.*, 2015) and the dominance selection mechanism. A solution i is said to constrainedly dominate a solution j if one of the following conditions holds:

(i) Solution i is feasible and Solution j is not.

(ii) Solutions i and j are feasible and Solution i dominates Solution j.

(iii) Solutions i and j are both infeasible, but Solution i has more satisfied constraints or has a smaller overall constraint violation when the two solutions have the same number of satisfied constraints.

The principle of sequence constrained domination is that any feasible solution has a better rank than any infeasible solution and all feasible solutions are ranked according to their nondomination level based on the objective function values. For two infeasible solutions, there are only two situations: (a) the solution with a larger number of satisfied constraints has a better rank, or (b) if both have the same number of satisfied constraints, the solution with a smaller overall constraint violation has a better rank. It should be noted that the objective functions are only evaluated if the solution satisfies all specified constraints in this feasibility-first scheme. Therefore, the computational efficiency can be improved.

4.6. Procedure of the SFMOFOA. By integrating the stochastic fractal based solution generation strategy, the Pareto dominance selection mechanism, the external elitist archive update scheme, the normalized nearest distance based crowding degree estimation, and the sequence constraint handling method, the computational procedure of the proposed SFMOFOA is outlined in Algorithm 2. According to the FOA procedure, the best individual in every iteration is chosen as the swarm location that is used to generate a new food source. Unfortunately, it is difficult to identify which solution is the best one for a MOP since all the nondominated solutions in a Pareto optimal solution set are identical in the sense of Pareto optimality. But when considering the spread of the nondominated solutions, it is desirable to have a good spread of the solutions which converge to

Algorithm 2. Stochastic fractal based multi-objective fruit fly optimization algorithm.

```
Require: NP, $T$, size of archive $A$
1: Initialize the population randomly in the search space
2: for $i = 1$ : NP do
3:     Evaluate the constraints and objective functions of $X_i(0)$
4:     Update the external elitist archive $A$
5: end for
6: Set the iteration counter $t = 1$
7: while $t < T$ or the stop criterion is not satisfied do
8:     Adaptively adjust $\delta$ according to (15)
9:     Choose the least crowded one as the candidate $X_b(t)$ from the elitist archive
10:    for $i = 1$ : NP do
11:        Generate a new solution $X_i^{'}(t)$ based on (14)
12:        Dominance or constraint-dominance selection between $X_i^{'}(t)$ and $X_i(t)$
13:        if $X_i^{'}(t)$ is is feasible and $X_i^{'}(t)$ dominate $X_i(t)$ or both are nondominated then
14:            Update the external elitist archive $A$
15:        end if
16:    end for
17:    while the size of archive > the given size do
18:        Calculate the normalized nearest neighbor distance
19:        Remove the most crowded one from the archive
20:    end while
21:    $t = t + 1$
22: end while
23: return The archive population $A$
```

the Pareto optimal set, and the least crowded individual in the external elitist archive can be chosen as the best individual. Therefore, the individual with the largest normalized nearest distance in the external elitist archive is always chosen and used to generate new solutions by the stochastic fractal. Obviously, the convergence and diversity are both highlighted by adopting the external elitist archive strategy and least crowded candidate approach.

5. Numerical analysis

In order to verify the performance of the proposed SFMOFOA, a total of eighteen benchmark functions are considered, including eleven bi-objective minimization problems: Schaffer, Kursawe, Fonseca, ConstrEx, Srinivas and Tanaka (Deb *et al.*, 2002), ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6 (Zitzler *et al.*, 2000), and seven tri-objective minimization problems: DTLZ1, DTZL2, DTZL2, DTLZ4, DTLZ5, DTLZ6 and DTLZ7 (Deb *et al.*, 2005). Among them, ConstrEx, Srinivas and Tanaka are constrained optimization problems. Optimization results from the SFMOFOA are compared with four state-of-the-art MOEAs, namely, NSGA-II (Deb *et al.*, 2002), SPEA2 (Zitzler *et al.*, 2001), MOPSO (Coello *et al.*, 2004), and MOSADE (Wang *et al.*, 2010). To avoid randomness, each function is optimized over 30 independent runs. For fair comparison, a total of 30 different initial populations were considered, starting from which each algorithm was run 30 times, and the relevant overall performance was then compared.

5.1. Quality indicators.

To demonstrate the effectiveness of an algorithm for solving MOPs, appropriate quality indicators should be chosen. There are three goals in multiobjective optimization: (i) convergence to the Pareto optimal set, (ii) maintenance of diversity in solutions of the Pareto optimal set and (iii) a maximal distribution bound of the Pareto optimal set. In this paper, the following three classic quality indicators are used to evaluate the performance of the MOEAs to be compared.

Generational distance (GD). The concept of generational distance was introduced by Van Veldhuizen and Lamont (Voss, 1991) to measure how far the elements are in the obtained set of nondominated vectors (P) from those in the true Pareto optimal set (P^*). It is defined as

$$\mathrm{GD} = \frac{1}{|P|}\sqrt{\sum_{X\in P} \mathrm{mindist}(X, P^*)^2}, \tag{19}$$

where mindist(X, P^*) is the minimum Euclidean distance (measured in the objective space) between a solution X and the solutions in P^*. A smaller value of the GD demonstrates a better convergence to the Pareto front.

Spread ($\triangle$): This indicator (Falconer, 1986; Wang *et al.*, 2010) is to measure the extent of spread archived among the obtained non-dominated solutions. A smaller value of $\triangle$ indicates a better distribution and diversity of the nondominated solutions. This indicator is defined as

$$\triangle = \frac{\sum_{i=1}^{m} d(E_i, \Omega) + \sum_{X\in\Omega} |d(X,\Omega) - \bar{d}|}{\sum_{i=1}^{m} d(E_i, \Omega) + (|\Omega| - m)\bar{d}}, \tag{20}$$

where Ω is a set of solutions, $(E_i, \dots, E_m)$ are m extreme solutions in the Pareto optimal set, m is the number of objectives and

$$d(X, \Omega) = \min_{Y\in\Omega, Y\neq X} ||F(X) - F(Y)||, \tag{21}$$

$$\bar{d} = \frac{1}{\Omega}\sum_{X\in\Omega} d(X, \Omega). \tag{22}$$

Hypervolume (HV). This indicator calculates the volume (in the objective space) covered by members of a nondominated set of solutions Ω for problems where all objectives are to be minimized (Coello, 2006). Mathematically, for each solution $X_i \in \Omega$, a hypercube v_i is constructed with a reference point W and the solution X_i as the diagonal corners of the hypercube. The reference point can be found simply by constructing a vector of worst objective function values. Thereafter, a union of all hypercubes is found and its hypervolume (HV) is calculated to be:

$$\text{HV} = \cup_{i=1}^{|\Omega|} v_i \tag{23}$$

Algorithms with larger HV values are desirable. Since the calculation of the HV is related to the reference point, in our experiments, the HV value of a set of solutions is normalized by a reference set of Pareto optimal solutions with the same reference point. After normalization, the HV values are confined to [0, 1].

5.2. Parameter setting. The population size for all the compared algorithms is NP $=$ 100, and all algorithms with an archive have the same archive size of 100 for fair comparison. The maximum generations was set to 250 for bi-objective MOPs, 500 for test problems with more than two objectives. For fair performance comparison, all the algorithms used equal maximum number of function evaluations ($\max_$NFEs). Other parameter settings of the compared algorithms are determined according to their original references. The source code of NSGA-II and MOPSO is available at `delta.cs.cinvestav.mx/~ccoello/EMOO/`. The implementation of the SPEA2 is available at `www.tik.ee.ethz.ch/pisa`.

5.3. Experimental results and comparisons. The results of the GD, $\triangle$ and HV indicators are presented in Tables 1, 2 and 3, respectively. The mean values and standard deviation values over all independent runs are calculated. The results in Table 1 are summarized as "w/l/n", which stands for "win/lose/non-convergence". For example, "14/1/3" in the fourth column means the SFMOFOA is significantly better than NSGA-II in 14 tested problems, but significantly worse in 1 tested problem, and NSGA-II did not converge in 3 tested problems. The results in Tables 2 and 3 are also summarized as "w/l", which means that the SFMOFOA is significantly better than and worse than the corresponding competitor on w and l problems, respectively. In order to facilitate observation, the best entries are marked in bold, and the non-convergence results are marked in italic.

As it can be seen from Table 1, the SFMOFOA obtains better GD results than NSGA-II, SPEA2 and MOPSO for all of the test problems except Srinivas. When it is compared with MOSADE, the SFMOFOA also obtains better results for all problems except Fonseca, Srinivas, ZDT4 and DTLZ7. It should be noticed that, for all problems, the SFMOFOA and MOSADE can converge to the optimal Pareto, while MOPSO cannot converge to the optimal Pareto for the seven problems of ZDT1, ZDT2, ZDT3, ZDT4, DLTZ1, DTLZ3 and DTLZ5, also, NSGA-II is not able to find the right Pareto optimal set for the problems of DTLZ2, DTLZ3 and DTLZ6, and SPEA2 cannot converge to the Pareto fronts for the problems of Kursawe, ZDT4, DTLZ1, DTLZ3, DTLZ5 and DTLZ7. That is to say, for the vast majority of the test problems, the SFMOFOA has better convergence performance when compared with MOSADE, NSGA-II, SPEA2 and MOPSO. To further demonstrate the performance of different algorithms, the graphical illustration of the nondominated solutions for the two problems of Kursawe and ZDT4 obtained by the SFMOFOA, NSGA-II, SPEA2, MOPSO and MOSADE in the objective space are shown in Figs. 2 and 3, respectively. It is clear from Figs. 2 and 3 that the SFMOFOA, NSGA-II and MOSADE can approximate the Pareto fronts of the two problems, but the SFMOFOA and MOSADE generally got better convergence performance than NSGA-II. However, SPEA2 is unable to approach the Pareto fronts for either of the two problems, and MOPSO cannot converge to the Pareto optimal solutions for the problem of ZDT4 in the current maximum function evaluations. Obviously, the graphical demonstrations are consistent with the statistic results in Table 1.

The results obtained from the spread indicator given in Table 2 show that the SFMOFOA is generally better than NSGA-II and MOPSO concerning the diversity of the obtained non-dominated solutions. MOSADE, however, with a crowding entropy-based diversity maintenance strategy, provides better spread values than the SFMOFOA for the bi-objective problems, but the SFMOFOA obtains six better values in the seven tri-objective problems (the set of DTLZ problems). That is to say, for the spread indicator, the SFMOFOA loses for the bi-objective problems but wins for the tri-objective problems compared with MOSADE. As the SFMOFOA adopts an improved diversity maintenance strategy based on the normalized nearest neighbor distance, it is expected that the SFMOFOA has better spread performance compared with SPEA2, especially for problems with Pareto front composed of multiple non-identical parts and different orders of magnitude of objective function values. But it is surprising from Table 2 that SPEA2 obtains eight better spread values than the SFMOFOA. However, when carefully observing Table 1 it can be found that the better spread of SPEA2 for four of them is based on the fact that those nondominated solutions obtained by SPEA2 are not well converged to the true Pareto fronts compared with those solutions given by the SFMOFOA.

Table 1. Comparison results of the SFMOFOA and other MOEAs based on the GD.

FUN	SFMOFOA	MOSADE	NSGA-II	SPEA2	MOPSO
Schaffer	**2.35E-04 ± 8.25E-06**	2.00E-03 ± 1.13E-04	2.16E-03 ± 2.10E-04	2.12E-03 ± 2.11E-04	2.93E-02 ± 1.56E-02
Kursaw	**2.39E-04 ± 1.78E-05**	2.44E-03 ± 2.64E-03	2.90E-03 ± 2.36E-04	*7.16E-01 ± 1.25E-02*	3.01E-02 ± 1.74E-03
Fonseca	1.26E-03 ± 1.85E-04	**1.24E-03 ± 7.07E-05**	2.57E-03 ± 2.01E-04	1.86E-03 ± 1.07E-04	2.14E-02 ± 7.42E-03
ConstrEx	**5.19E-04 ± 2.23E-05**	4.81E-03 ± 3.74E-04	5.13E-03 ± 2.48E-04	4.82E-03 ± 3.77E-04	4.54E-03 ± 6.86E-04
Srinivas	4.43E-02 ± 1.08E-02	**2.00E-03 ± 2.02E-04**	3.71E-03 ± 5.10E-04	2.11E-03 ± 4.75E-04	2.76E-03 ± 2.08E-04
Tanaka	**6.35E-04 ± 1.19E-04**	3.74E-03 ± 3.79E-04	4.05E-03 ± 4.35E-04	3.82E-03 ± 4.91E-04	5.09E-03 ± 4.56E-04
ZDT1	**2.06E-04 ± 5.57E-05**	1.25E-03 ± 9.76E-05	1.34E-03 ± 1.41E-04	8.61E-03 ± 2.60E-03	*1.86E-01 ± 7.74E-02*
ZDT2	**1.57E-04 ± 2.94E-05**	9.81E-04 ± 4.91E-05	9.81E-04 ± 6.41E-04	2.48E-02 ± 1.61E-02	*5.24E-01 ± 2.97E-01*
ZDT3	**2.61E-04 ± 3.61E-05**	2.16E-03 ± 2.00E-04	2.48E-03 ± 1.27E-04	9.72E-03 ± 5.23E-03	*4.34E-01 ± 6.49E-02*
ZDT4	2.73E-03 ± 1.59E-03	**1.20E-03 ± 8.37E-05**	5.16E-02 ± 1.33E-03	*9.25E-01 ± 4.28E-01*	-
ZDT6	**7.87E-04 ± 3.62E-05**	2.62E-03 ± 1.10E-04	6.08E-03 ± 6.08E-03	1.93E-02 ± 1.40E-03	5.21E-02 ± 2.50E-02
DTLZ1	**4.94E-04 ± 1.88E-05**	5.25E-03 ± 1.57E-04	2.71E-02 ± 6.65E-02	*9.03E-01 ± 7.69E-01*	*2.14E-01 ± 1.56E-01*
DTLZ2	**5.49E-04 ± 1.63E-05**	5.33E-03 ± 1.74E-04	*1.02E-01 ± 1.06E-01*	2.81E-02 ± 1.25E-02	9.13E-02 ± 2.69E-02
DTLZ3	**5.82E-04 ± 6.14E-05**	6.46E-03 ± 3.33E-04	*5.09E-01 ± 8.10E-01*	*8.13E-01 ± 1.22E+00*	-
DTLZ4	**5.68E-04 ± 3.93E-05**	4.77E-03 ± 2.74E-04	1.06E-02 ± 7.98E-02	1.73E-02 ± 9.91E-02	4.26E-02 ± 1.33E-02
DTLZ5	**4.60E-04 ± 2.19E-05**	3.77E-03 ± 3.15E-04	4.80E-03 ± 4.04E-04	*4.76E-01 ± 1.22E-01*	-
DTLZ6	**4.47E-04 ± 2.54E-05**	3.89E-03 ± 2.01E-04	*4.54E-01 ± 2.02E-04*	4.30E-03 ± 4.41E-04	1.41E-02 ± 8.77E-02
DTLZ7	3.38E-03 ± 2.42E-04	**2.37E-03 ± 1.30E-03**	2.81E-02 ± 1.82E-02	*1.49E-01 ± 6.93E-02*	9.88E-02 ± 3.61E-03
$w/l/n$	-/-/-	14/1/3	14/1/3	11/1/6	10/1/7

Table 2. Comparison results of the SFMOFOA and other MOEAs based on △.

FUN	SFMOFOA	MOSADE	NSGA-II	SPEA2	MOPSO
Schaffer	1.79E-01 ± 9.62E-03	**1.35E-01 ± 1.88E-02**	2.92E-01 ± 2.13E-02	2.75E-01 ± 2.57E-02	7.26E-01 ± 1.35E-01
Kursaw	1.67E-01 ± 1.38E-02	**2.82E-01 ± 2.10E-02**	4.34E-01 ± 1.98E-02	*2.86E-01 ± 1.15E-02*	3.76E-01 ± 1.67E-02
Fonseca	3.11E-01 ± 1.25E-02	**1.16E-01 ± 6.57E-03**	3.77E-01 ± 2.52E-02	1.77E-01 ± 1.11E-01	*6.50E-01 ± 3.12E-01*
ConstrEx	**3.04E-01 ± 1.57E-02**	3.55E-01 ± 5.80E-02	5.49E-01 ± 2.72E-02	4.28E-01 ± 1.23E-02	9.43E-01 ± 3.67E-01
Srinivas	2.06E-01 ± 1.40E-02	**1.01E-01 ± 9.38E-03**	3.87E-01 ± 2.51E-02	1.01E-01 ± 1.91E-02	6.66E-01 ± 7.22E-02
Tanaka	9.89E-01 ± 3.58E-02	**7.53E-01 ± 2.99E-02**	8.23E-01 ± 2.87E-02	7.84E-01 ± 3.00E-02	7.94E-01 ± 5.10E-02
ZDT1	3.29E-01 ± 2.78E-02	**1.32E-01 ± 5.69E-03**	5.04E-01 ± 3.93E-02	2.96E-01 ± 1.09E-01	*2.94E-01 ± 1.70E-02*
ZDT2	3.23E-01 ± 4.95E-02	**1.21E-01 ± 7.94E-03**	4.88E-01 ± 2.77E-02	5.05E-01 ± 1.84E-01	*2.88E-01 ± 1.76E-02*
ZDT3	9.00E-01 ± 6.26E-02	**4.38E-01 ± 8.08E-03**	5.90E-01 ± 3.04E-02	5.03E-01 ± 9.73E-02	*6.18E-01 ± 3.50E-02*
ZDT4	9.45E-01 ± 9.09E-02	**1.18E-01 ± 5.86E-03**	3.75E-01 ± 2.44E-02	*7.28E-01 ± 5.15E-01*	3.24E-01 ± 3.30E-02
ZDT6	2.41E-01 ± 2.82E-02	**1.33E-01 ± 9.83E-03**	4.86E-01 ± 3.61E-02	2.49E-01 ± 4.97E-02	1.12E+00 ± 1.73E-01
DTLZ1	**2.10E-01 ± 2.07E-02**	6.38E-01 ± 3.05E-02	9.19E-01 ± 6.61E-02	*3.02E-01 ± 2.49E-01*	*7.08E-01 ± 7.89E-02*
DTLZ2	**2.05E-01 ± 2.24E-02**	6.13E-01 ± 3.67E-02	*8.31E-01 ± 6.84E-02*	2.47E-01 ± 3.52E-02	9.36E-01 ± 3.17E-01
DTLZ3	**2.16E-01 ± 2.11E-02**	5.86E-01 ± 3.00E-02	*9.56E-01 ± 1.78E-01*	*3.18E-01 ± 2.37E-01*	-
DTLZ4	**2.30E-01 ± 2.63E-02**	8.32E-01 ± 1.09E-01	7.79E-01 ± 7.81E-02	2.56E-01 ± 7.56E-02	1.50E+00 ± 3.31E-01
DTLZ5	3.45E-01 ± 4.24E-02	4.24E-01 ± 2.21E-02	7.11E-01 ± 7.80E-02	***2.98E-01 ± 1.84E-02***	5.41E-01 ± 5.58E-01
DTLZ6	**3.64E-01 ± 3.52E-02**	4.26E-01 ± 4.75E-02	*1.09E+00 ± 1.83E-01*	5.08E-01 ± 9.22E-02	-
DTLZ7	4.97E-01 ± 4.36E-02	8.55E-01 ± 3.86E-02	1.01E+00 ± 5.06E-02	***3.98E-01 ± 7.03E-02***	8.22E-01 ± 8.97E-02
w/l	-/-	8/10	17/1	10/8	17/1

To further investigate the aforementioned phenomenon, the non-dominated solutions for the problems of ConstrEx and DTLZ2 obtained by the five algorithms in the objective space are shown in Figs. 4 and 5, respectively. As can be seen from Fig. 4, the Pareto front for ConstrEx is composed of two parts. The left part covers the vast majority of the range of the second objective function values and the slope is very steep in the shape. In contrast, the right part is very flat and only covers a very limited range of the second objective function values. Moreover, the two parts have different orders of magnitude of objective function values. It can be observed that the non-dominated solutions obtained by the SFMOFOA are nearly uniformly distributed along the whole Pareto front as the objective functions were normalized prior to the calculation of the distances of

Table 3. Comparison results of the SFMOFOA and other MOEAs based on the HV.

FUN	SFMOFOA	MOSADE	NSGA-II	SPEA2	MOPSO
Schaffer	**9.97E-01** ± **5.27E-05**	9.97E-01 ± 4.46E-05	9.97E-01 ± 1.04E-04	9.97E-01 ± 1.13E-04	9.96E-01 ± 2.10E-03
Kursaw	**9.95E-01** ± **6.83E-04**	9.89E-01 ± 2.45E-03	7.86E-01 ± 3.63E-04	*8.24E-01* ± *2.20E-04*	9.89E-01 ± 1.10E-03
Fonseca	**9.98E-01** ± **8.14E-04**	9.88E-01 ± 1.59E-04	9.80E-01 ± 1.03E+01	9.85E-01 ± 4.72E-04	9.55E-01 ± 2.07E-02
ConstrEx	9.93E-01 ± 6.61E-04	**9.95E-01** ± **1.97E-04**	9.92E-01 ± 5.85E-04	9.94E-01 ± 6.59E-04	9.83E-01 ± 1.29E-02
Srinivas	**9.96E-01** ± **4.66E-03**	9.94E-01 ± 4.45E-04	9.93E-01 ± 2.29E-04	9.94E-01 ± 1.47E-03	9.89E-01 ± 1.69E-03
Tanaka	9.85E-01 ± 3.26E-03	9.96E-01 ± 6.18E-04	**9.98E-01** ± **9.28E-04**	9.96E-01 ± 9.38E-04	9.87E-01 ± 3.78E-03
ZDT1	**9.97E-01** ± **2.74E-04**	**9.96E-01** ± **4.41E-05**	9.95E-01 ± 1.89E-04	9.91E-01 ± 6.33E-04	*8.21E-01* ± *7.05E-02*
ZDT2	9.96E-01 ± 3.38E-04	**9.97E-01** ± **4.01E-05**	9.96E-01 ± 2.20E-04	9.87E-01 ± 2.71E-03	*6.03E-01* ± *1.62E-01*
ZDT3	9.97E-01 ± 7.90E-04	**9.99E-01** ± **3.98E-05**	9.93E-01 ± 9.81E-03	9.83E-01 ± 7.11E-03	*6.27E-01* ± *4.49E-02*
ZDT4	9.72E-01 ± 1.09E-02	**9.98E-01** ± **3.16E-06**	9.90E-01 ± 1.80E-03	*6.46E-01* ± *1.35E-01*	*6.33E-01* ± *2.11E-01*
ZDT6	**9.99E+00** ± **1.94E-04**	9.99E-01 ± 8.83E-05	8.72E-01 ± 1.02E-02	9.39E-01 ± 4.38E-03	9.95E-01 ± 2.67E-02
DTLZ1	**9.95E-01** ± 1.50E-03	9.93E-01 ± 5.52E-05	9.92E-01 ± 3.26E-04	*9.64E-01* ± *4.36E-02*	*9.95E-01* ± *4.73E-03*
DTLZ2	9.70E-01 ± 2.62E-03	9.97E-01 ± 1.52E-04	*9.95E-01* ± *1.02E-03*	9.96E-01 ± 3.30E-04	9.73E-01 ± 4.62E-02
DTLZ3	9.65E-01 ± 1.57E-03	**9.97E-01** ± **1.54E-04**	*9.11E-01* ± *9.01E-02*	*9.10E-01* ± *8.06E-02*	*0.00E+00* ± *0.00E+00*
DTLZ4	**9.98E-01** ± **7.66E-04**	9.97E-01 ± 3.77E-04	9.76E-01 ± 3.11E-02	9.77E-01 ± 3.18E-02	9.93E-01 ± 1.37E-03
DTLZ5	**9.95E-01** ± **1.27E-03**	9.89E-01 ± 2.02E-05	9.87E-01 ± 1.03E-03	*0.00E+00* ± *0.00E+00*	*0.00E+00* ± *0.00E+00*
DTLZ6	9.93E-01 ± 3.67E-05	9.88E-01 ± 1.52E-05	*0.00E+00* ± *0.00E+00*	**9.99E-01** ± **1.37E-04**	9.94E-01 ± 9.08E-03
DTLZ7	**9.76E-01** ± **3.67E-05**	9.69E-01 ± 4.23E-03	9.63E-01 ± 4.03E-03	*9.56E-01* ± *4.45E-03*	9.32E-01 ± 5.75E-03
w/l	-/-	10/8	14/4	14/4	15/3

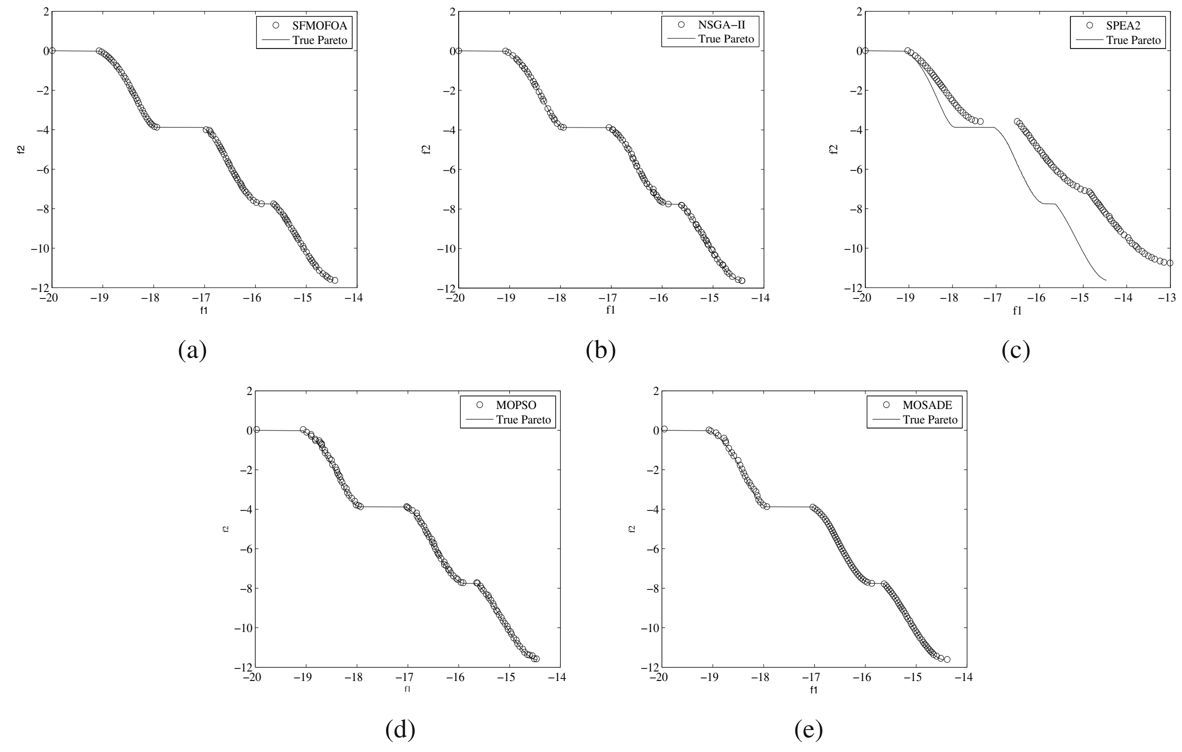

Fig. 2. Pareto fronts obtained by the five algorithms on the Kursawe problem.

between solutions, and therefore the Pareto front shape of the two parts are nearly identical. When it comes to SPEA2, as expected, almost all solutions are distributed on the left part while the right part has few solutions. NSGA-II and MOSADE (using a crowding distance based diversity strategies) and MOPSO (using an adaptive hypercube based strategy) produce similar results as SPEA2 for the problem of ConstrEx. Moreover, the result for ConstrEx also confirms the effectiveness of the proposed constraint handling strategy in dealing with

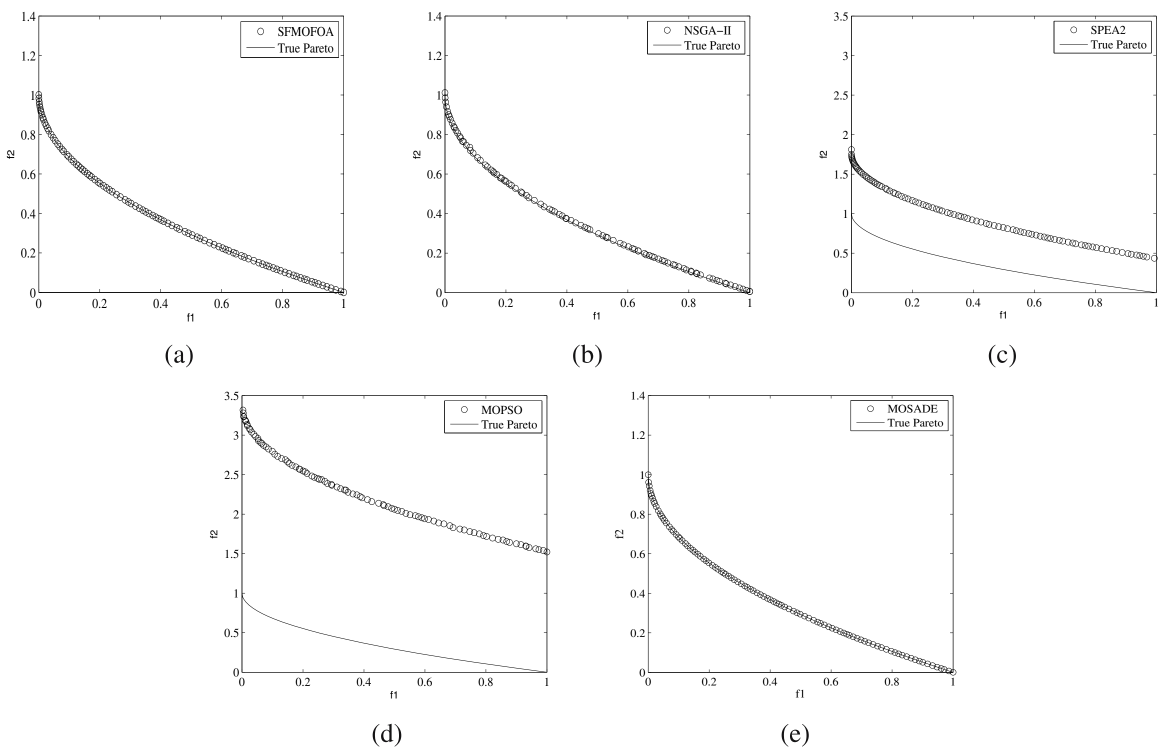

Fig. 3. Pareto fronts obtained by the five algorithms on the ZDT4 problem.

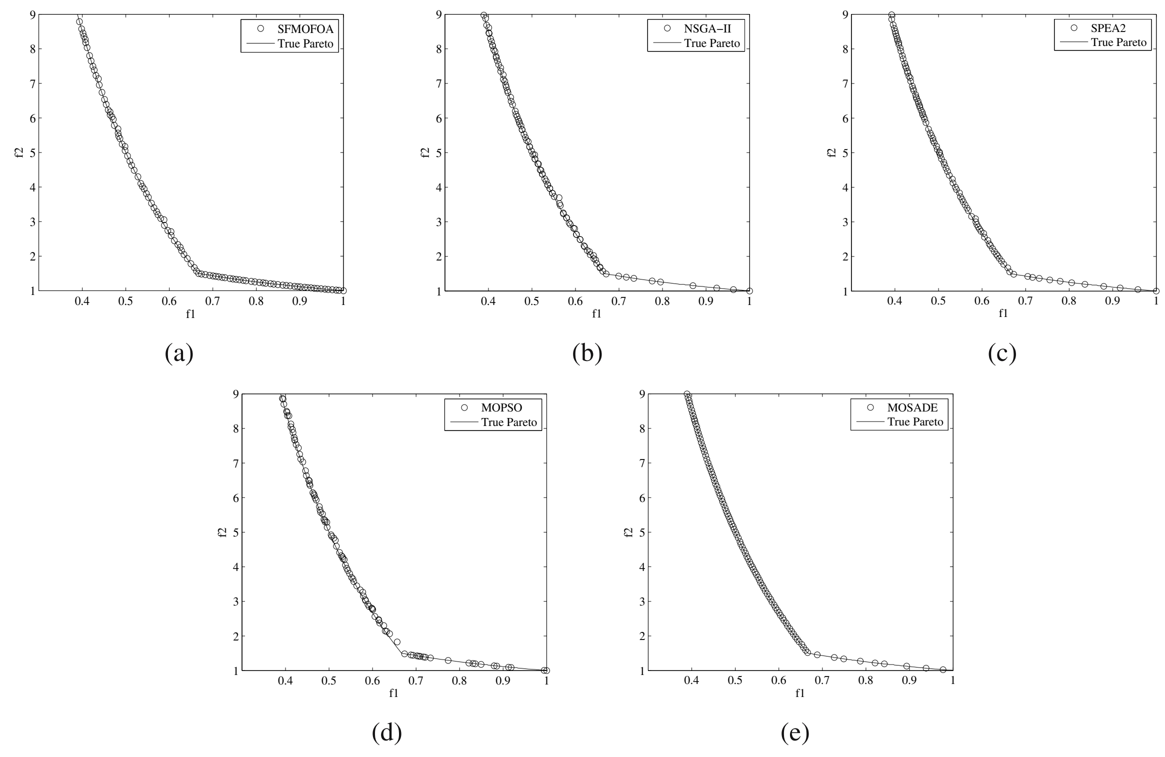

Fig. 4. Pareto fronts obtained by the five algorithms on the ConstrEx problem.

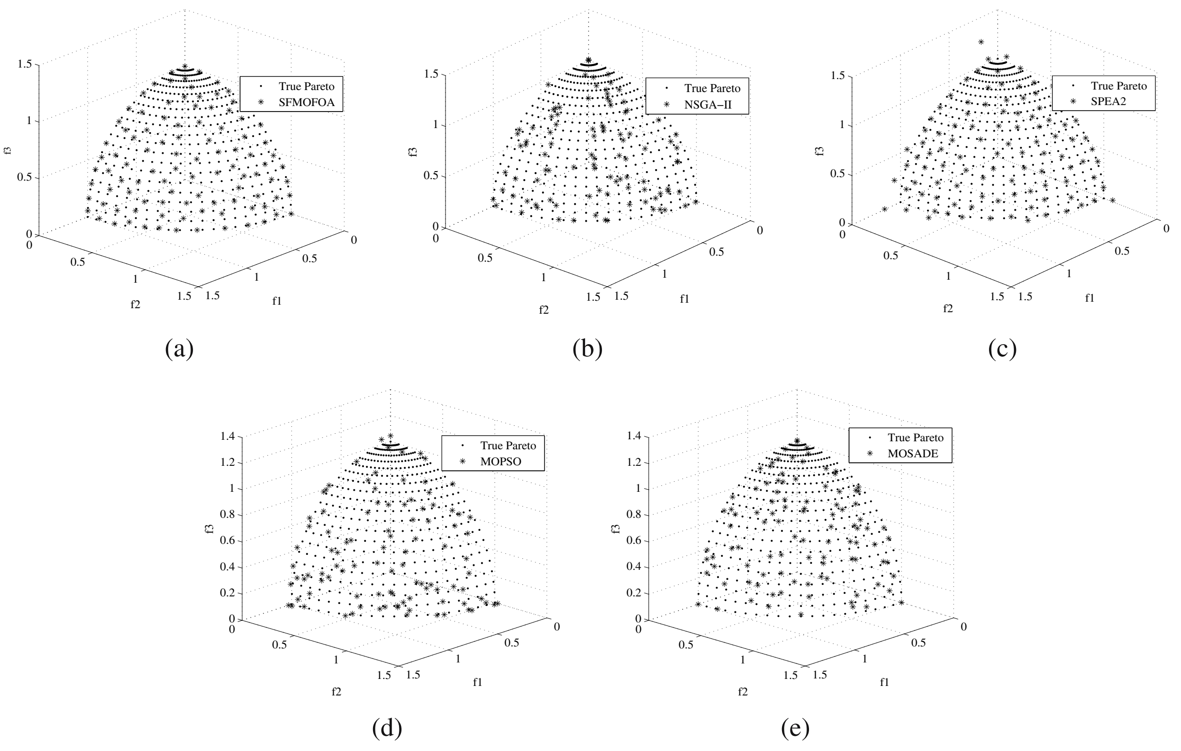

Fig. 5. Pareto fronts obtained by the five algorithms on the DTLZ2 problem.

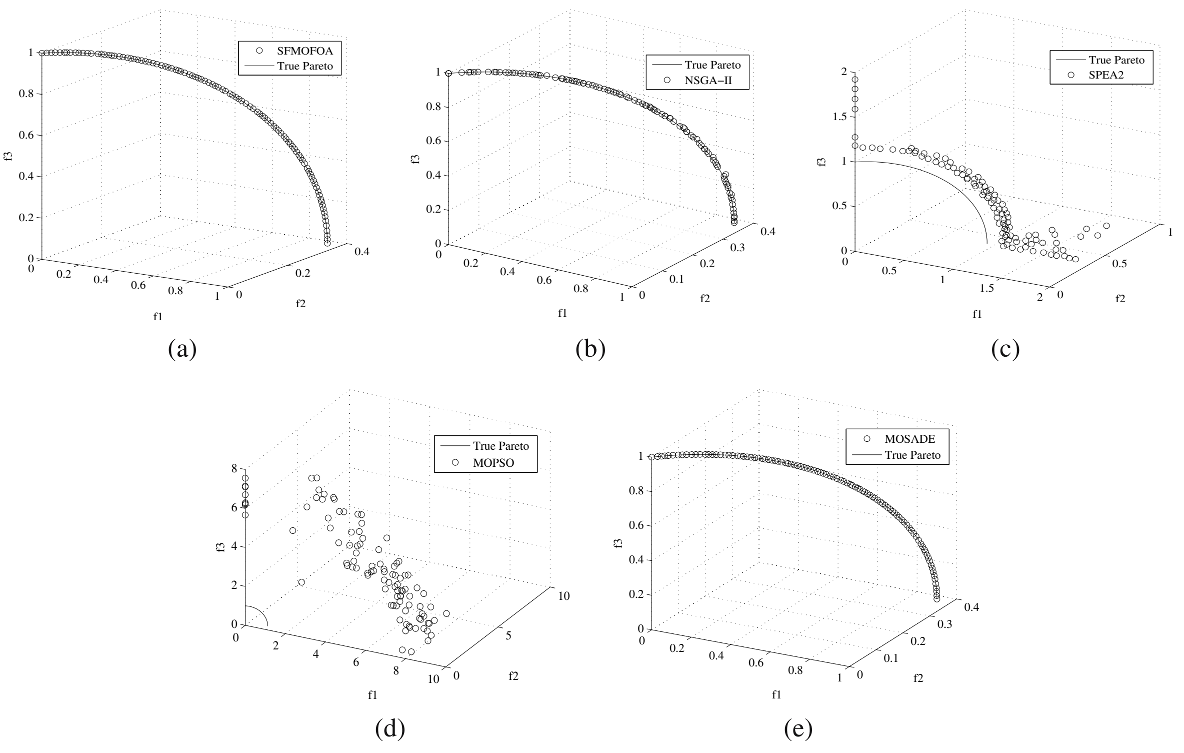

Fig. 6. Pareto fronts obtained by the five algorithms on the DTLZ5 problem.

the constrained MOP. For problems with more than two objective functions, as can be seen from Fig. 4 that the SFMOFOA and SPEA2 have better spread performance than NSGA-II, MOPSO and MOSADE. Therefore, it can be concluded that crowding distance based strategies are good choices for bi-objective problems while the nearest neighbor distance based methods are more effective for problems with three or more objectives. What is also worth noticing is that some of the solutions obtained by SPEA2 do not converge to the true Pareto front, which can be clearly observed from Fig. 5.

As a measure of both convergence and diversity, the hypervolume (HV) may be used to test the results of the other two indicators. As can be seen from Table 3, the SFMOFOA produces ten larger HV values than MOSADE. This means that the SFMOFOA is competitive to MOSADE in both convergence and diversity performance for these test problems. With a further inspection of Table 3, the SFMOFOA produces 14, 14, and 15 larger HV values when comparing with NSGA-II, SPEA2, and MOPSO, respectively. These results indicate that SFMOFOA has better convergence and diversity performance than NSGA-II, SPEA2 and MOPSO for most of the test benchmarks. Take DTLZ5 as an example, the HV value generated by SPEA2 and MOPSO is zero, meaning that the Pareto fronts obtained by the two algorithms are outside the limits of the true Pareto front of the problem. To graphically demonstrate this fact, the nondominated solutions to the problem DTLZ5 obtained by the SFMOFOA, NSGA-II, SPEA2, MOPSO and MOSADE (in the objective space) are shown in Fig. 6.

The above analyses clearly show that the SFMOFOA can generate a significantly better convergence performance than NSGA-II, SPEA2 and MOPSO and better than MOSADE in most of cases for the 18 test problems. The good convergence performance mainly benefits from the effective local search capability of the stochastic fractal with parameter adaptive updating strategy. The local search method can find a very promising convergence to the true Pareto front compared with the other strategies (Chen *et al.*, 2015). Moreover, the SFMOFOA significantly outperforms NSGA-II, MOPSO and SPEA2 and is competitive with MOSADE in diversity performance, which results from the normalized nearest neighbor distance-based density estimator.

To further testify the above conclusions, the Wilcoxon signed-rank test was carried out and the results for the three quality indicators for all Benchmark problems are proposed in Table 4. The Wilcoxon signed ranks test is a simple yet safe and robust nonparametric test for pairwise statistical comparisons (Derrac *et al.*, 2011). It is a pairwise test that aims to detect significant differences between two sample means. The results of the Wilcoxon signed-rank test were calculated by the statistical software tool KEEL (Alcalá-Fdez *et al.*, 2009).

From Table 4, it is clear that the SFMOFOA obtained higher R^+ values than R^- values in the GD indicator, which means that the SFMOFOA significantly outperforms NSGA-II, SPEA2, MOPSO and MOSADE in convergence performance. For the spread indicator Δ, the SFMOFOA is significantly different from NSGA-II and MOPSO at 5% level of significance. That is to say, the SFMOFOA has a significantly better performance of diversity maintenance than NSGA-II and MOPSO. But compared with MOSADE and SPEA2, the difference is not significant (the p-values are approximately equal to 0.5). The main reason is that the SFMOFOA loses in bi-objective problems but wins in tri-objective problems in the Δ indicator. When considering the HV indicator, the SFMOFOA outperforms NSGA-II, SPEA2 and MOPSO at 5% level of significance, but has a comparable performance with MOSADE. Obviously, the multi-problem Wilcoxon signed-rank test results are consistent with the above conclusions from Tables 1–3.

6. Conclusions

An improved novel fruit fly optimization algorithm based on stochastic fractals for multiobjective optimization was proposed. To avoid the limitations of the original definition of the smell concentration judgement in the FOA and improve its convergence performance, a new position update mechanism based on the Gaussian walk based fractal growth was developed. In addition, an adaptive parameter strategy was introduced to dynamically adjust the search range according to the current swarm location and evolution process to balance the exploration and exploitation. All this significantly improved the search power of fruit fly swarm, both locally and globally, as now it enables a more effective search in a relatively larger space.

Based on the Pareto dominance concept, a dominance selection operator was developed and the improved FOA was then extended for multiobjective optimization problems. Similarly to many other MOEAs, an external elitist archive was utilized to preserve the non-dominated solutions found so far during the evolution. For the purpose of finding a set of Pareto optimal solutions that are uniformly distributed along the whole Pareto front, the normalized nearest neighbor distance based density estimation strategy was proposed to keep the size of the archive. The normalized nearest neighbor distance can accurately reflect the crowding degree of solutions in the archive, especially for problems with the Pareto front composed of multiple nonidentical parts and different orders of magnitude of objective function values. In addition, the concept of sequence constrained-domination is proposed to handle constrained MOPs by means of an improved feasibility-first scheme

Table 4. Wilcoxon signed-rank result for the SFMOFOA vs. MOSADE, NSGA-II, SPEA2 and MOPSO.

SFMOFOA	GD			Δ			HV		
vs.	R^+	R^-	p-value	R^+	R^-	p-value	R^+	R^-	p-value
MOSADE	144.0	27.0	8.96E-3	81.0	90.0	5.37E-1	77.5	75.5	4.85E-1
NSGA-II	158.0	13.0	6.71E-4	141.0	30.0	1.38E-2	134.5	36.5	3.23E-2
SPEA2	159.0	12.0	5.34E-4	103.0	68.0	4.68E-1	131.5	21.5	7.27E-3
MOPSO	164.0	7.0	1.45E-4	146.0	25.0	6.32E-3	144.5	8.5	4.43E-4

and associate dominance selection mechanism. It follows from the analysis results proposed in Section 5 that overall the proposed SFMOFOA outperforms the four state-of-the-art methods, namely, NSGA-II, SPEA2, MOPSO and MOSADE.

A potential drawback of the proposed SFMOFOA is that its performance is sensitive to the adaptive strategy of δ in the Gaussian walk. The choice of the power α in Eqn. (16) may affect the performance of the algorithm for various MOPs. For the future work, a more robust adaptive strategy should be further considered. Moreover, we are planning to extend the proposed SFMOFOA to dynamic multiobjective optimization problems and practical applications, e.g., to environmental systems modelling and economic dispatch of power systems.

Acknowledgment

This work was partly supported by the National Natural Science Foundation of PR China (grants no. 61203309, 61403134, 61672226), the Engineering and Physical Sciences Research Council (EPSRC) under the grant EP/I011056/1 and the Platform grant EP/H00453X/1, the ERC Horizon 2020 Research and Innovation Action Framework Programme under the grant no. 637302 (PROGRESS), and the Hunan Provincial Innovation Foundation for Postgraduates (CX2015B488).

References

Aguirre, A.H., Rionda, S.B., Coello Coello, C.A., Lizárraga, G.L. and Montes, E.M. (2004). Handling constraints using multiobjective optimization concepts, *International Journal for Numerical Methods in Engineering* **59**(15): 1989–2017.

Alcalá-Fdez, J., Sanchez, L., Garcia, S., del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J. and Rivas, V.M. (2009). KEEL: A software tool to assess evolutionary algorithms for data mining problems, *Soft Computing* **13**(3): 307–318.

Asafuddoula, M., Ray, T. and Sarker, R. (2015). A differential evolution algorithm with constraint sequencing: An efficient approach for problems with inequality constraints, *Applied Soft Computing* **36**: 101–113.

Awad, N.H., Ali, M.Z., Suganthan, P.N. and Jaser, E. (2016). A decremental stochastic fractal differential evolution for global numerical optimization, *Information Sciences* **372**: 470–491.

Bader, J. and Zitzler, E. (2011). Hype: An algorithm for fast hypervolume-based many-objective optimization, *Evolutionary computation* **19**(1): 45–76.

Ben Aicha, F., Bouani, F. and Ksouri, M. (2013). A multivariable multiobjective predictive controller, *International Journal of Applied Mathematics and Computer Science* **23**(1): 35–45, DOI: 10.2478/amcs-2013-0004.

Bilski, P. and Wojciechowski, J. (2014). Artificial intelligence methods in diagnostics of analog systems, *International Journal of Applied Mathematics and Computer Science* **24**(2): 271–282, DOI: 10.2478/amcs-2014-0020.

Bonabeau, E., Dorigo, M. and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, Oxford.

Chen, B., Zeng, W., Lin, Y. and Zhang, D. (2015). A new local search-based multiobjective optimization algorithm, *IEEE Transactions on Evolutionary Computation* **19**(1): 50–73.

Chen, P.-W., Lin, W.-Y., Huang, T.-H. and Pan, W.-T. (2013). Using fruit fly optimization algorithm optimized grey model neural network to perform satisfaction analysis for e-business service, *Applied Mathematics & Information Sciences* **7**(2L): 459–465.

Cheng, R., Jin, Y., Narukawa, K. and Sendhoff, B. (2015). A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling, *IEEE Transactions on Evolutionary Computation* **19**(6): 838–856.

Coello, C.A.C., Pulido, G.T. and Lechuga, M.S. (2004). Handling multiple objectives with particle swarm optimization, *IEEE Transactions on Evolutionary Computation* **8**(3): 256–279.

Coello, C.C. (2006). Evolutionary multi-objective optimization: A historical view of the field, *IEEE Computational Intelligence Magazine* **1**(1): 28–36.

Deb, K. (2000). An efficient constraint handling method for genetic algorithms, *Computer Methods in Applied Mechanics and Engineering* **186**(2): 311–338.

Deb, K. and Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach. Part I: Solving problems with box constraints, *IEEE Transactions on Evolutionary Computation* **18**(4): 577–601.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation* **6**(2): 182–197.

Deb, K., Thiele, L., Laumanns, M. and Zitzler, E. (2005). *Scalable Test Problems for Evolutionary Multiobjective Optimization*, Springer, London.

Denysiuk, R., Costa, L., Santo, I.E. and Matos, J.C. (2015). MOEA/PC: Multiobjective evolutionary algorithm based on polar coordinates, *International Conference on Evolutionary Multi-Criterion Optimization, Guimarães, Portugal*, pp. 141–155.

Derrac, J., García, S., Molina, D. and Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm and Evolutionary Computation* **1**(1): 3–18.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics* **26**(1): 29–41.

Eiben, A.E. and Smith, J. (2015). From evolutionary computation to the evolution of things, *Nature* **521**(7553): 476–482.

Falconer, K.J. (1986). *Random Fractals*, Cambridge University Press, Cambridge, pp. 559–582.

Fonseca, C.M. and Fleming, P.J. (1998). Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I: A unified formulation, *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans* **28**(1): 26–37.

Ishibuchi, H., Doi, T. and Nojima, Y. (2006). Incorporation of scalarizing fitness functions into evolutionary multiobjective optimization algorithms, *in* T.P. Runarsson *et al.* (Eds.), *Parallel Problem Solving from Nature, PPSN IX*, Springer, Berlin/Heidelberg, pp. 493–502.

Karaboga, D. and Akay, B. (2009). A comparative study of artificial bee colony algorithm, *Applied Mathematics and Computation* **214**(1): 108–132.

Kennedy, J. (2011). Particle swarm optimization, *in* C. Sammut and G.I. Webb (Eds.), *Encyclopedia of Machine Learning*, Springer, Berlin/Heidelberg, pp. 760–766.

Li, C., Xu, S., Li, W. and Hu, L. (2012). A novel modified fly optimization algorithm for designing the self-tuning proportional integral derivative controller, *Journal of Convergence Information Technology* **7**(16): 69–77.

Li, H.-Z., Guo, S., Li, C.-J. and Sun, J.-Q. (2013). A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm, *Knowledge-Based Systems* **37**(2): 378–387.

Lin, S.-M. (2013). Analysis of service satisfaction in web auction logistics service using a combination of fruit fly optimization algorithm and general regression neural network, *Neural Computing and Applications* **22**(3–4): 783–791.

Liu, H.-L., Gu, F. and Zhang, Q. (2014). Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems, *IEEE Transactions on Evolutionary Computation* **18**(3): 450–455.

Mandelbrot, B.B. (1983). *The Fractal Geometry of Nature*, Macmillan, New York, NY.

Mei, Y., Tang, K. and Yao, X. (2011). Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem, *IEEE Transactions on Evolutionary Computation* **15**(2): 151–165.

Mellal, M.A. and Zio, E. (2016). A penalty guided stochastic fractal search approach for system reliability optimization, *Reliability Engineering & System Safety* **152**: 213–227.

Menchaca-Mendez, A. and Coello, C. A.C. (2015). GD-MOEA: A new multi-objective evolutionary algorithm based on the generational distance indicator, *International Conference on Evolutionary Multi-Criterion Optimization, Guimarães, Portugal*, pp. 156–170.

Michalewicz, Z. and Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimization problems, *Evolutionary Computation* **4**(1): 1–32.

Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014). Grey wolf optimizer, *Advances in Engineering Software* **69**: 46–61.

Mitić, M., Vuković, N., Petrović, M. and Miljković, Z. (2015). Chaotic fruit fly optimization algorithm, *Knowledge-Based Systems* **89**(C): 446–458.

Mousavi, S.M., Alikar, N. and Niaki, S.T.A. (2016). An improved fruit fly optimization algorithm to solve the homogeneous fuzzy series-parallel redundancy allocation problem under discount-strategies, *Soft Computing* **20**(6): 2281–2307.

Niu, J., Zhong, W., Liang, Y., Luo, N. and Qian, F. (2015). Fruit fly optimization algorithm based on differential evolution and its application on gasification process operation optimization, *Knowledge-Based Systems* **88**(C): 253–263.

Pan, Q.-K., Sang, H.-Y., Duan, J.-H. and Gao, L. (2014). An improved fruit fly optimization algorithm for continuous function optimization problems, *Knowledge-Based Systems* **62**(5): 69–83.

Pan, W. (2011). A new evolutionary computation approach: Fruit fly optimization algorithm, *Proceedings of the Conference on Digital Technology and Innovation Management, Taipei, Taiwan*.

Pan, W.-T. (2012). A new fruit fly optimization algorithm: Taking the financial distress model as an example, *Knowledge-Based Systems* **26**(2): 69–74.

Pan, W.-T. (2013). Using modified fruit fly optimisation algorithm to perform the function test and case studies, *Connection Science* **25**(2–3): 151–160.

Rafajłowicz, E. and Rafajłowicz, W. (2012). Fletcher's filter methodology as a soft selector in evolutionary algorithms for constrained optimization, *in* L. Rutkowski *et al.* (Eds.), *Swarm and Evolutionary Computation*, Springer, Berlin/Heidelberg, pp. 333–341.

Rafajłowicz, W. (2013). Method of handling constraints in differential evolution using Fletcher's filter, *Proceedings of the 12th International Conference on Artificial Intelligence and Soft Computing (ICAICS 2013), Zakopane, Poland*, pp. 46–55.

Rajabioun, R. (2011). Cuckoo optimization algorithm, *Applied Soft Computing* **11**(8): 5508–5518.

Rodríguez Villalobos, C.A. and Coello Coello, C.A. (2012). A new multi-objective evolutionary algorithm based on a performance assessment indicator, *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation, Philadelphia, PA, USA*, pp. 505–512.

Salimi, H. (2015). Stochastic fractal search: A powerful metaheuristic algorithm, *Knowledge-Based Systems* **75**(C): 1–18.

Shan, D., Cao, G. and Dong, H. (2013). LGMS-FOA: An improved fruit fly optimization algorithm for solving optimization problems, *Mathematical Problems in Engineering* **2013**(7): 1256–1271.

Shen, L., Chen, H., Yu, Z., Kang, W., Zhang, B., Li, H., Yang, B. and Liu, D. (2016). Evolving support vector machines using fruit fly optimization for medical data classification, *Knowledge-Based Systems* **96**(C): 61–75.

Sheng, W. and Bao, Y. (2013). Fruit fly optimization algorithm based fractional order fuzzy-PID controller for electronic throttle, *Nonlinear Dynamics* **73**(1–2): 611–619.

Van Veldhuizen, D.A. and Lamont, G.B. (1998). Multiobjective evolutionary algorithm research: A history and analysis, *Technical Report TR-98-03*, Air Force Institute of Technology, Wright-Patterson AFB, OH.

Vicsek, T. and Gould, H. (1989). Fractal growth phenomena, *Computers in Physics* **3**(5): 108.

Voss, R.F. (1991). Random fractals: Characterization and measurement, *in* R. Pynn and A. Skjeltrop (Eds.), *Scaling Phenomena in Disordered Systems*, Springer, New York, NY, pp. 1–11.

Wang, L., Shi, Y. and Liu, S. (2015). An improved fruit fly optimization algorithm and its application to joint replenishment problems, *Expert Systems with Applications* **42**(9): 4310–4323.

Wang, L., Zheng, X.-l. and Wang, S.-Y. (2013). A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem, *Knowledge-Based Systems* **48**(2): 17–23.

Wang, R., Zhang, Q. and Zhang, T. (2015). Pareto adaptive scalarising functions for decomposition based algorithms, *International Conference on Evolutionary Multi-Criterion Optimization, Guimarães, Portugal*, pp. 248–262.

Wang, Y.-N., Wu, L.-H. and Yuan, X.-F. (2010). Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure, *Soft Computing* **14**(3): 193–209.

Witten, T.A. and Sander, L.M. (1983). Diffusion-limited aggregation, *Physical Review B* **27**(9): 5686.

Wu, L., Zuo, C. and Zhang, H. (2015a). A cloud model based fruit fly optimization algorithm, *Knowledge-Based Systems* **89**(C): 603–617.

Wu, L., Zuo, C., Zhang, H. and Liu, Z. (2015b). Bimodal fruit fly optimization algorithm based on cloud model learning, *Soft Computing* **21**(7): 1877–1893.

Yamaguchi, M., Hata, M. and Kigami, J. (1997). *Mathematics of Fractals*, American Mathematical Society, New York, NY.

Yang, X.-S. (2010). Firefly algorithm, stochastic test functions and design optimisation, *International Journal of Bio-Inspired Computation* **2**(2): 78–84.

Yuan, X., Dai, X., Zhao, J. and He, Q. (2014). On a novel multi-swarm fruit fly optimization algorithm and its application, *Applied Mathematics and Computation* **233**(3): 260–271.

Yuan, X., Liu, Y., Xiang, Y. and Yan, X. (2015). Parameter identification of BIPT system using chaotic-enhanced fruit fly optimization algorithm, *Applied Mathematics and Computation* **268**(C): 1267–1281.

Zhang, Q. and Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Transactions on Evolutionary Computation* **11**(6): 712–731.

Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P.N. and Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art, *Swarm and Evolutionary Computation* **1**(1): 32–49.

Zitzler, E., Deb, K. and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results, *Evolutionary Computation* **8**(2): 173–195.

Zitzler, E. and Künzli, S. (2004). Indicator-based selection in multiobjective search, *International Conference on Parallel Problem Solving from Nature, Birmingham, UK*, pp. 832–842.

Zitzler, E., Laumanns, M. and Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm, *Eurogen* **3242**: 95–100.

Cili Zuo received the BSc degree in electrical engineering and automation from the Hunan University of Science and Technology, Xiangtan, China, in 2012, where he currently works toward the MSc degree with the School of Information and Electrical Engineering. His research interests include intelligent optimization algorithm, multi-objective optimization, and their applications.

Lianghong Wu received the BSc degree in industrial automation from the Hunan University of Science and Technology, Xiangtan, China, in 2001, and the MSc and PhD degrees in control science and engineering from Hunan University, Changsha, China, in 2007 and 2011, respectively. From 2015 to 2016, he was a visiting scientist with the Department of Automatic Control and Systems Engineering, University of Sheffield. He is currently a professor with the School of Information and Electrical Engineering, Hunan University of Science and Technology. His research interests include artificial intelligence and evolutionary computation, and their applications.

Zhao-Fu Zeng received the BSc degree in industrial automation from the Hunan University of Science and Technology, Xiangtan, China, in 1991, and the MSc degrees in communication engineering from the National Defense University of Science and Technology, Changsha, China, in 2002. Currently, he is pursuing his PhD from Central South University. His research interests include artificial intelligence and evolutionary computation, and their applications.

Hua-Liang Wei received his BSc and MSc in China and his PhD from the University of Sheffield, Sheffield, UK, in 2004. He previously held academic appointments (lecturer and associate professorship) from 1992 to 2000 at the Beijing Institute of Technology, Beijing, China. He joined the Department of Automatic Control and Systems Engineering, University of Sheffield, UK, in 2004. Dr. Wei is the head of the Laboratories of Dynamical Modelling, Data Mining and Decision Making (3DM) and Digital Medicine & Computational Neuroscience (DMCN), University of Sheffield. His recent research interests include dynamic modelling and system identification, NARMAX methodology and its applications, machine learning, non-stationary processes tracking and characterization, linear and nonlinear optimization, with multidisciplinary applications in medicine and biomedicine, medical informatics, space weather, environmental and social sciences. He is currently serving as the editor-in-chief of the *International Journal of Signal and Imaging Systems Engineering*.

Received: 5 October 2016
Revised: 10 February 2017
Accepted: 8 March 2017