Applied Mathematics and Computation 233 (2014) 260–271

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

On a novel multi-swarm fruit fly optimization algorithm and its application ☆

Xiaofang Yuan [a,*], Xiangshan Dai [a], Jingyi Zhao [a], Qian He [b]

[a] College of Electrical and Information Engineering, Hunan University, Changsha 410082, China
[b] School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China

ARTICLE INFO

Keywords:
Optimization algorithm
Fruit fly optimization algorithm (FOA)
Multi-swarm
Swarm behavior
Cooperative swarms

ABSTRACT

Swarm intelligence is a research field that models the collective behavior in swarms of insects or animals. Recently, a kind of Drosophila (fruit fly) inspired optimization algorithm, called fruit fly optimization algorithm (FOA), has been developed. This paper presents a variation on original FOA technique, named multi-swarm fruit fly optimization algorithm (MFOA), employing multi-swarm behavior to significantly improve the performance. In the MFOA approach, several sub-swarms moving independently in the search space with the aim of simultaneously exploring global optimal at the same time, and local behavior between sub-swarms are also considered. In addition, several other improvements for original FOA technique is also considered, such as: shrunk exploring radius using osphresis, and a new distance function. Application of the proposed MFOA approach on several benchmark functions and parameter identification of synchronous generator shows an effective improvement in its performance over original FOA technique.

1. Introduction

In the last few decades, more and more researches suggest that nature is a great source for inspirations to both develop intelligent systems and provide solutions to complicated problems [1]. Taking animals for example, evolutionary pressure forces them to develop highly optimized organs and skills to take advantages of fighting for food, territories and mates. Some of the organs and skills can be well refined as optimization algorithms, and the evolution is a process to fine-tune the parameter settings in the algorithms [1].

The collective intelligent behavior of insect or animal groups in nature such as flocks of birds, colonies of ants, schools of fish, swarms of bees and termites have attracted the attention of researchers [2]. In recent years, some examples of swarm behavior inspired optimization algorithms are particle swarm optimization (PSO) algorithm [3,4], ant colony optimization (ACO) [5,6], artificial bee colony (ABC) algorithm [7,8], social spider optimization (SSO) [2], artificial fish swarm algorithm [9], firefly algorithm (FA) [10]. Simulations and applications have shown that these swarm behavior inspired meta-heuristics in [2–10] have good search capability as well as potential applications.

☆ This work was supported in part by the National Natural Science Foundation of China (Nos. 61104088, 61201250) and Young Teachers Promotion Program of Hunan University.

* Corresponding author.
E-mail address: yuanxiaofang@hnu.edu.cn (X. Yuan).

http://dx.doi.org/10.1016/j.amc.2014.02.005
0096-3003/

One current trend in swarm behavior research is focusing more on the Drosophila (fruit fly) species [11,12]. Recently, a kind of Drosophila inspired optimization algorithm has been developed, called fruit fly optimization algorithm (FOA) [13,14], which is a novel evolutionary computation and optimization technique. The FOA is a new approach for finding global optimization based on the swarm behavior of fruit fly. The main inspiration of FOA is that the fruit fly itself is superior to other species in sensing and perception, especially in osphresis and vision. The FOA technique has the advantages of being easy to understand and to be written into program code which is not too long. More recently, FOA technique has been applied in several applications, such as swarms of mini autonomous surface vehicles [13], neural network parameters optimization [14–17], PID controller parameters tuning [18], key control characteristics optimization [19] etc. In order to improve the search efficiency and global search ability, several researchers have also presented improved FOA [19–22]. However, for these FOAs, fruit fly swarm will use vision to fly towards so far best smell position, this implies that fruit flies will be around the so far best smell position at a fast speed. The diversity loss occurs when the global optimal is shifted away from a too converged swarm. This kind of swarm behavior is quite similar to being trapped in local optimal or premature in multi-model optimization problems.

This paper presents a variation on the original FOA technique, named multi-swarm fruit fly optimization algorithm (MFOA), employing multi-swarms behavior to significantly improve the performance. In the MFOA approach, several sub-swarms moving independently in the search space with the aim of simultaneously exploring global optimal at the same time, and local behavior between sub-swarms are also considered. In addition, several other improvements for original FOA technique is also considered, such as: shrunk exploring radius using osphresis, and a new distance function. Application of this new MFOA approach on several benchmark functions and parameter identification of synchronous generator shows a marked improvement in performance over original FOA technique.

The rest of this paper is organized as follows. Review of FOA technique is summarized in Section 2. Section 3 describes the motivation and implement of the MFOA approach in detail. In Section 4, the testing of the proposed MFOA approach through benchmark problems and parameter identification of synchronous generator is carried out and the simulation results are compared. Finally, the conclusion is drawn in Section 5.

2. FOA technique

2.1. Swarm behavior of fruit fly

The osphresis organs of fruit fly can find all kinds of scents floating in the air; it can even smell food source from 40 km away. Then, after it gets close to the food location, it can also use its sensitive vision to find food and the company's flocking location, and fly towards that direction too [13]. When a fly decides to go for hunting, it will fly randomly to find the location guided by a particular odor. While searching, a fly also sends and receives information from its neighbors and makes comparison about the so far best location and fitness [13]. If a fly has found its favorable spot, it will then identify the fitness by taste. If the location no longer exists or the taste is 'bitter', the fly will go off searching again. The fly will stay around at the most profitable area, sending, receiving and comparing information with its swarm at the same time [13].

The main idea behind the FOA technique is based upon the Drosophila's biological behavior [13]: (1) The fly flies with Levy flight motion; (2) It smells the potential location (attractiveness); (3) It would then taste. If it is not to its liking (fitness/profitability), it rejects and goes to another location. To the fly, attractiveness is not necessarily profitable; (4) While foraging or mating, the fly also sends and receives messages with its swarm about its food and their mates.

2.2. FOA technique implementation

Based on the food finding characteristics of fruit fly swarm, a kind of FOA technique is proposed in [13,14], which is a novel evolutionary computation and optimization technique. Although the FOAs are inspired by swarm behavior of fruit fly, the implement procedure of FOA in [13] is different from that in [14–18]. This can be considered as two different ways of computing implement for FOA technique. In this paper, we will focus on FOA technique in [14–18] for improvement. The FOA in [14–18] can be divided into several necessary steps and the main steps are described as follows:

Step 1. Random initial fruit fly swarm location as shown in Fig. 1. *Init X_axis*; *Init Y_axis*.

Step 2. Give the random direction and distance for the search of food using osphresis by an individual fruit fly.

$$\begin{aligned} X_i &= X\_axis + RandomValue \\ Y_i &= Y\_axis + RandomValue \end{aligned} \tag{1}$$

where i is the population size of fruit flies.

Step 3. Since the food location cannot be known, the distance to the origin is thus estimated first (*Dist*), then the smell concentration judgment value (S) is calculated, and this value is the reciprocal of distance.

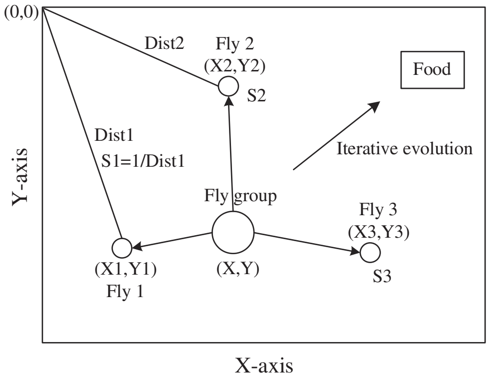

Fig. 1. The group iterative food searching of fruit flies.

$$Dist_i = \sqrt{X_i^2 + Y_i^2}$$
$$S_i = \frac{1}{Dist_i} \quad (2)$$

Step 4. Substitute smell concentration judgment value (S) into smell concentration judgment function (or called Fitness function) so as to find the smell concentration ($Smell_i$) of the individual location of the fruit fly.

$$Smell_i = Function(S_i) \quad (3)$$

Step 5. Find out the fruit fly with maximal smell concentration (finding the maximal value) among the fruit fly swarm.

$$[bestSmell\ bestIndex] = max(Smell) \quad (4)$$

Step 6. Keep the best smell concentration value and x, y coordinate, and at this moment, the fruit fly swarm will use vision to fly towards that location.

$$Smellbest = bestSmell$$
$$X\_axis = X(bestIndex)$$
$$Y\_axis = Y(bestIndex) \quad (5)$$

Step 7. Enter iterative optimization to repeat the implementation of **Step 2**–**Step 5**, then judge if the smell concentration is superior to the previous iterative smell concentration, if so, implement **Step 6**.

3. MFOA approach

3.1. Motivation of MFOA

Based on the simulations and analysis, the steps of original FOA technique in searching the optimal solution would reveal that it does not have high probability of mutation; hence, the searching space will be limited, and it is difficult for FOA technique to jump away from the local extremum [20]. So in this section, some improvements for original FOA are pointed out as follows:

(1) Multi-swarm strategy. In **Step 6** of original FOA, the fruit fly swarm will use vision to fly towards that location, this implies that fruit flies will be around the so far best position (X_axis, Y_axis) at a fast speed. The diversity loss occurs when the global optimal is shifted away from a too converged swarm. This behavior is quite similar to being trapped in local optimal or premature in multi-model optimization problems.
Instead of having one swarm trying to find the optimal solution, in this paper, the swarm is split into several sub-swarms (usually 4 to 10) so that sub-swarms move independently in the search space with the aim of simultaneously exploring global optimal at the same time. Usually, the sub-swarm has the same size of population. This approach use several sub-swarms mainly in order to enhance the diversity of solutions and achieve an effective exploration to avoid local optimal or premature. In the MFOA, cooperative sub-swarms are also employed, this is also a kind of local search. This local search is used as the supplement of MFOA around the so far best values, and it may not guide the MFOA to local optimum.

(2) Revised valuation function. In fact, on the basis of analysis and computations of the feature values $Dist_i$ and S_i in Eq. (2), it is obvious that numerical values of $Dist_i$ are distributed randomly in large-scale scopes. However, the large scope of $Dist_i$ numerical values dealt with by $S_i = 1/Dist_i$ in Eq. (2), causes that the scope of S_i becomes very small. When S_i is

substituted into fitness function in Eq. (3), this directly causes the possibility of the premature convergence of FOA and makes FOA get into easily local optimal [18]. In this paper, both the distance $Dist_i$ and smell concentration judgement value S_i are removed, and the fitness function are evaluated based on the decision variable $X = (x_1, x_2, \ldots, x_n)$ directly as:

$$Smell_i = Function(X_i) \tag{6}$$

(3) Shrinking of exploring radius using osphresis. In **Step 2** of FOA, random direction and distance for the search of food using osphresis by an individual fruit fly is shown in Eq. (1). This also means that the osphresis is used for exploring in a random way. In this paper, multi-scale exploring radius using osphresis is employed, that is:

$$X_i = X\_axis + R(k) * RandomValue \tag{7}$$

where $R(k)$ is the exploring radius using osphresis, and this variable is set as a multi-scale factor according to the iteration times k. The shrinking of $R(k)$ is done at a variable rate. By this way, the exploring radius using osphresis will decrease as the increase of iteration times k. This approach may be useful for searching the optimum around the so far best points.

Based on the above analysis, the MFOA can be implemented in the following section.

3.2. Implement of MFOA

In this section, our aim is to present a novel MFOA approach based on multi-swarm strategy. The implement procedure of the proposed MFOA is illustrated in Fig. 2 and it is summarized as follows.

Consider the optimization problem for nonlinear function with boundary constraints:

$$min f(X) = f(x_1, x_2, \ldots, x_n), \quad x_j \in [a_j, b_j]. \tag{8}$$

where n is the number of decision variable, and j represents each decision variable.

Step 1. Initialization. Set the max iteration times k_{max}, let $k = 1$, population size of fruit flies $Popsize$, and sub-swarms number M. Initialize fruit fly swarm location $Init\ X\_axis$. In the following, $j = 1, 2, \ldots, n$ which represents each decision variable, $i = 1, 2, \ldots, Popsize$ which represents each population of fruit flies, $m = 1, 2, \ldots, M$ which represents each sub-swarm.

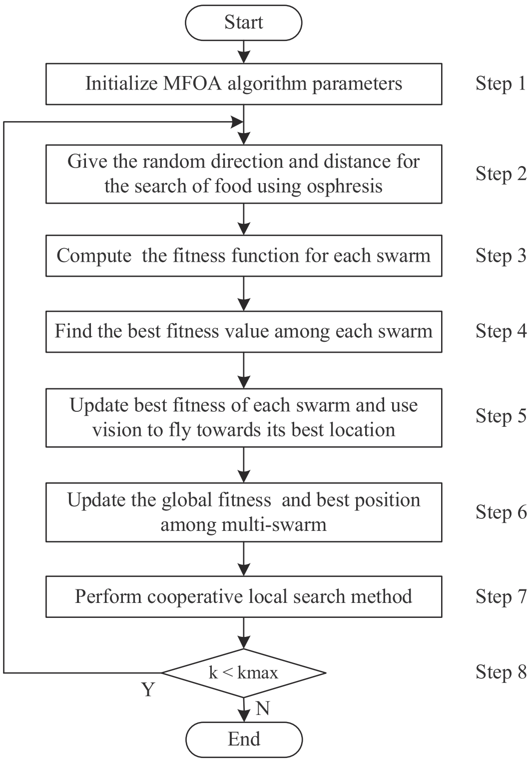

Fig. 2. The implement procedure of the proposed MFOA.

Step 2. Give the random direction and distance for the search of food using osphresis by an individual fruit fly, each swarm is conducted independently as:

$$X_{i,m} = X\_axis_m + R(k) * RandomValue \tag{9}$$

with

$$R(k) = \left(\frac{b_j - a_j}{2}\right) * \left(\frac{k_{max} - k}{k_{max}}\right)^{\phi} \tag{10}$$

where $\phi = 2 \sim 6$. Big $R(k)$ value in early iterations may increase the diversity of solution vectors for global exploration, while in final iterations small $R(k)$ value may enhance the fine-tuning of solution vectors by local exploitation.

Step 3. Substitute decision variable value (X_i) into fitness function or objective function so as to find the fitness function value ($Smell_i$) of the individual location of fruit fly.

$$Smell_i = Function(X_i) \tag{11}$$

Step 4. Find out the fruit fly with the minimum value or the best fitness value among each sub-swarm.

$$[bestSmell_m \ bestIndex_m] = min(Smell) \tag{12}$$

Step 5. Judge if the fitness of each sub-swarm is superior to the previous iterative fitness, if so, update the best fitness value of each sub-swarm, and at this moment, each sub-swarm will use vision to fly independently towards that location.

$$\begin{aligned} &Smellbest_m = bestSmell_m \\ &X\_axis_m = X(bestIndex_m) \end{aligned} \tag{13}$$

Step 6. Update the global fitness *Smellbest* and best position X_axis among multi-swarm by: If $Smellbest_m < Smellbest$, $Smellbest = Smellbest_m$, $X\_axis = X\_axis_m$.

Step 7. Perform cooperative local search method by

$$X\_new = \frac{1}{M}\sum_{m=1}^{M} X\_axis_m \tag{14}$$

If $Function(X\_new) < Smellbest$, update the global fitness and best position as: $Smellbest = Function(X\_new)$, $X\_axis = X\_new$.

Step 8. If $k \geqslant k_{max}$, stop the MFOA search; otherwise, go to **Step 2**.

In **Step 2**–**Step 5**, each sub-swarm searches its best fitness value $Smellbest_m$ and best position X_axis_m independently, and **Step 6** will search the global fitness *Smellbest* and best position X_axis among multi-swarm. In the manner of multi-swarm strategy, several sub-swarms can achieve an effective exploration of the search space to avoid local optimal or premature. In the proposed approach, each sub-swarm has the same number of population for easy implementation. Since both exploration and exploitation are stressed and balanced, it is expected to achieve good performances for optimization problems with limited buffers. In the next section, we will investigate the performance of the MFOA based on computational simulation and comparisons.

4. Simulation

4.1. Benchmark function performance

The efficiency and performance of the MFOA approach with the following six nonlinear functions [23,24] is evaluated:

$$f_1(x,y) = \left(4 - 2.1x^2 + \frac{x^4}{3}\right)x^2 + xy + (-4 + 4y^2)y^2, \quad -20 < x,y < 20 \tag{15}$$

$$f_2(x,y) = 0.5 - \frac{sin^2\sqrt{x^2+y^2} - 0.5}{(1 + 0.001(x^2+y^2))^2}, \quad -20 < x,y < 20 \tag{16}$$

$$f_3(x_i) = \sum_{i=1}^{3}(x_i^2 - 10cos(2\pi x_i) + 10), \quad -5 < x_i < 5, \ i = 1,2,3 \tag{17}$$

$$f_4(x_i) = \frac{1}{4000}\sum_{i=1}^{15}x_i^2 - \prod_{i=1}^{15}cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad -10 < x_i < 10, \ i = 1,2,\ldots,15 \tag{18}$$

$$f_5(x_i) = \frac{1}{15}\sum_{i=1}^{15}(x_i^4 - 16x_i^2 - 5x_i), \quad -10 < x_i < 10, \ i = 1,2,\ldots,15 \tag{19}$$

$$f_6(x_i) = \frac{1}{15}\sum_{i=1}^{15}\left(10(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right), \quad -10 < x_i < 10,\ i = 1, 2, \ldots, 15 \tag{20}$$

Function f_1 is the Camel function, which has six local minima and two global minima $x^* = (-0.0898, 0.7126), x^* = (0.0898, -0.7126)$, and optimal objective function value $f^* = -1.03162845$. Function f_2 is the Schaffer's function, which has infinite local maxima and one global maximum $x^* = (0, 0)$, and $f^* = 1.0$. Function f_3 is the Rastrigin's function, which has many local minima and one global minimum $x^* = (0, 0, 0)$, and $f^* = 0$. Function f_4 is the Griewank's function, which has many local minima and one global minimum $x^* = (0, 0, \ldots, 0)$, and $f^* = 0$. Function f_5 has 15 variables, which has 32 local minima and one global minimum $x^* = (2.9051, 2.9051, \ldots, 2.9051)$, and $f^* = -78.332314$. Function f_6 is the Rosenbrock's function with 15 variables, which has several local minima and one global minimum $x^* = (1, 1, \ldots, 1)$, and $f^* = 0$. These six nonlinear multi-modal functions are often used to test the convergence, efficiency and accuracy of optimization algorithms. Among them the former three functions have two or three design variables, which is 2- or 3-dimensional problem, and the latter three have 15 variables which is 15-dimensional problem.

In this simulation, population size $Popsize = 30$, multi swarm number $M = 5$. Fig. 3 shows fruit flies distribution at different iterations of MFOA optimization process for f1. Each sub-swarm with the same population size is denoted by different shapes. As the increase of iterations, $k = 5, 50, 100, 200$, the sub-swarms become more gathered, and they have reached two global optimum when $k = 200$.

Fig. 4 also illustrates fruit flies distribution at different iterations of MFOA optimization process for f2, where $Popsize = 30, M = 5, k = 5, 50, 100, 200$. In Fig. 4, one sub-swarm has reached a local maxima and it cannot jump to global

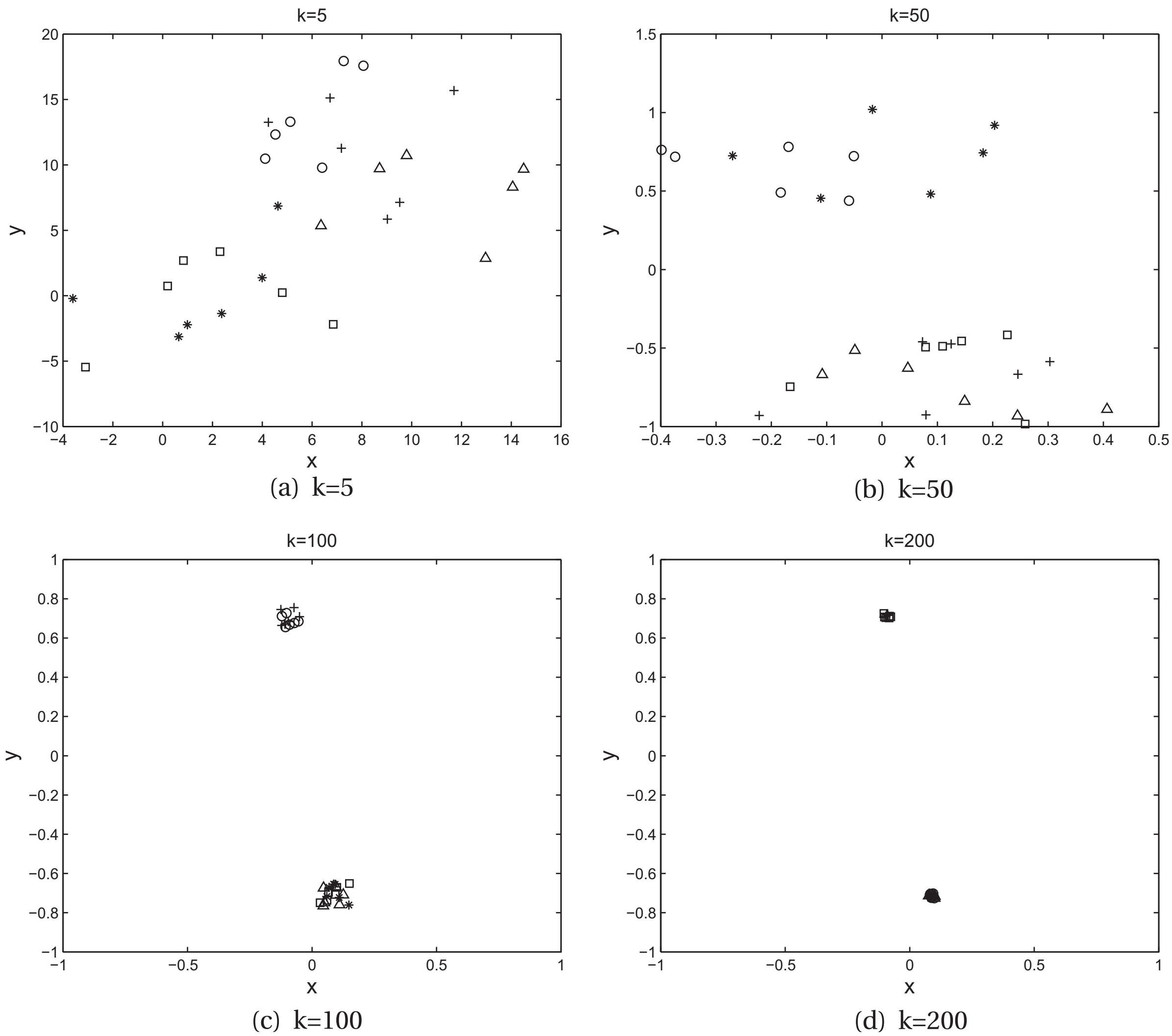

Fig. 3. Fruit flies distribution at different iterations for f1.

maxima when $k = 200$, while other sub-swarms have reached global maxima. It can be seen from Figs. 3 and 4 that the MFOA can more easily jump out local optimum by the multi-swarms strategy which may enhance the diversity of optimal solutions.

The best fruit fly flying routes for f1–f6 in this simulation by the proposed MFOA approach are illustrated in Fig. 5, which shows that the best fruit fly can direct to the global optimal solutions in an efficient way.

In Fig. 6, a typical optimization process for f1–f6 using both FOA and MFOA approach is illustrated. It can be seen from Fig. 6 that the MFOA approach can usually reach the global optimal fitness values faster than that of FOA. This also indicates that the MFOA outperforms FOA.

In order to show the optimization performance, the MFOA is compared with FOA for f1–f6, and the simulation results are reported in Fig. 7. The 'Best' means the optimal objective function value, the 'Worst' means the worst objective function value, and the 'Mean' means the average objective function value in 20 times searching for each algorithm; the 'Rate' means the successful rate in 20 times defined as: $\|g^* - X^*\| \leqslant (U - L) \times 10^{-4}$, where g^* and X^* are the global optimal solution and the obtained optimal solution by the optimization algorithm, respectively; U and L are upper and lower limits. In Fig. 7, the worst value of the FOA is usually local optimal, such as 0.990281(f2), 0.994962(f3), 0.105843(f4), −67.023039(f5), 0.497471(f6), while the MFOA has few local optimal in 20 times searching: −72.678150(f5), 0.494948(f6). According to the test results of these six nonlinear functions, MFOA approach has a better performance than original FOA with respect to index 'Best', 'Worst', 'Mean' and 'Rate'.

Here the proposed MFOA approach is also compared with the performances of four widely used evolutionary algorithm: particle swarm optimization (PSO) algorithm [25], covariance matrix adaptation evolution strategy (CMAES) [26], self-adaptive differential evolution (SaDE1) algorithm [27], and self-adaptive differential evolution (SaDE2) algorithm [28].

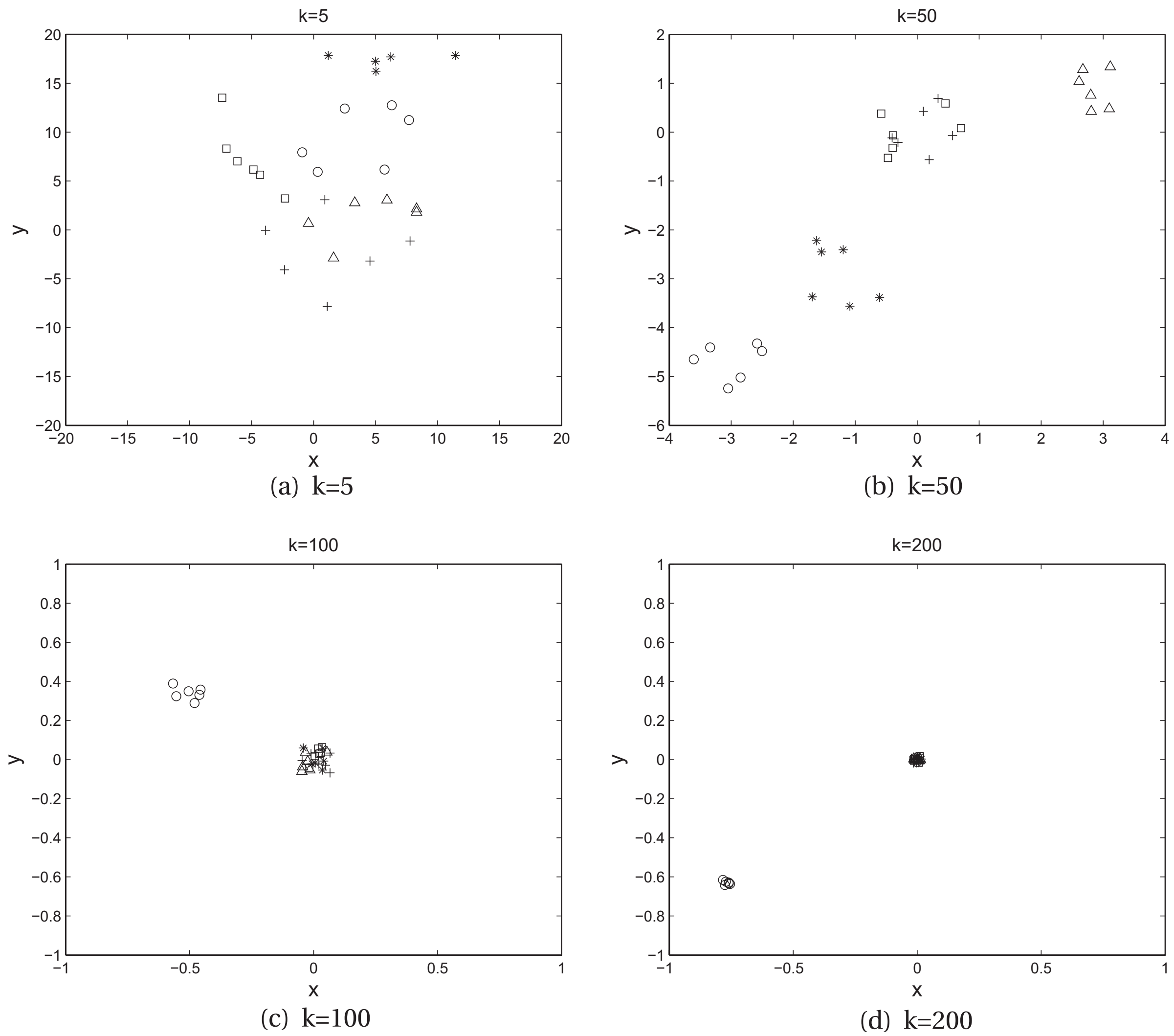

Fig. 4. Fruit flies distribution at different iterations for f2.

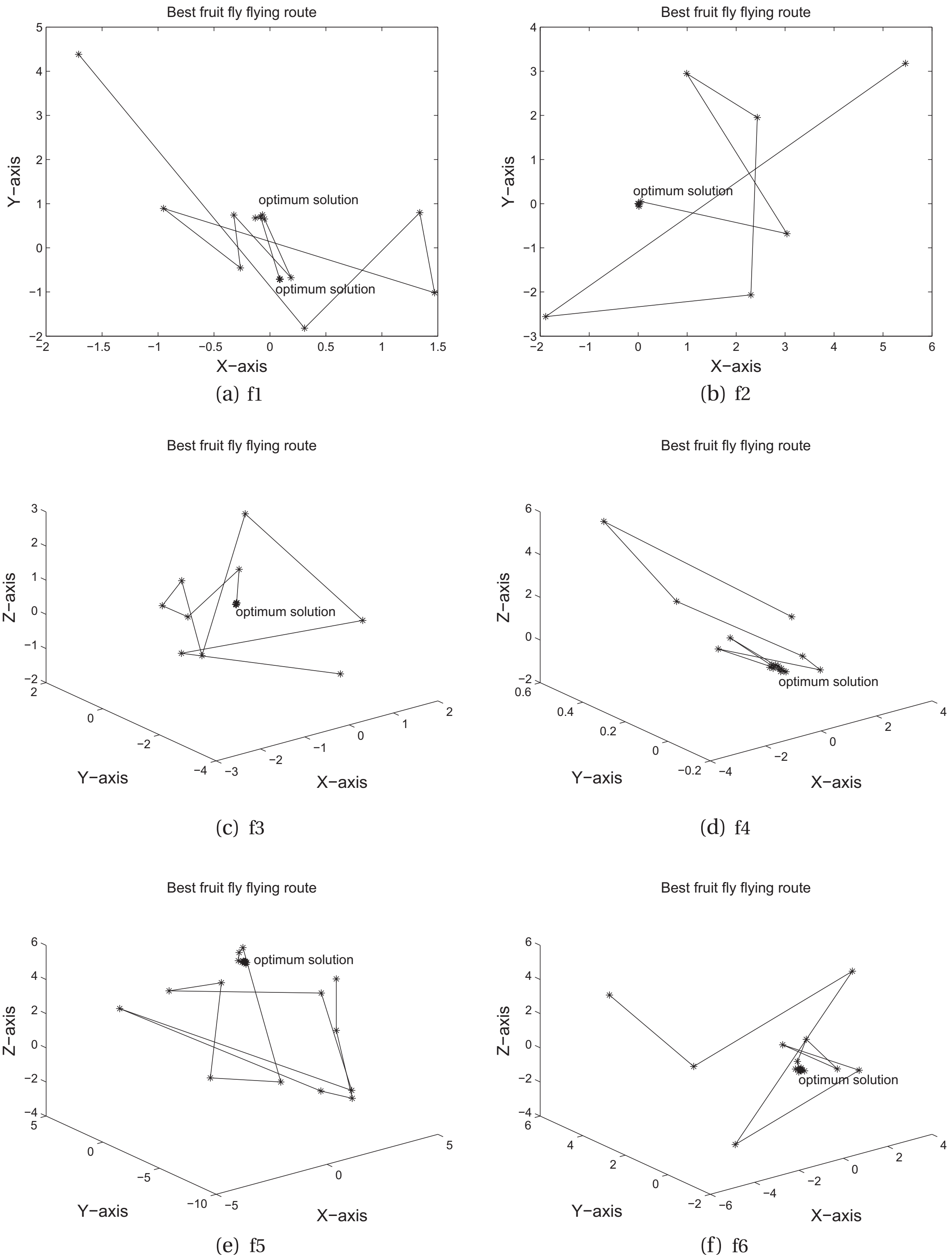

Fig. 5. Best fruit fly flying route for f1–f6 using MFOA approach.

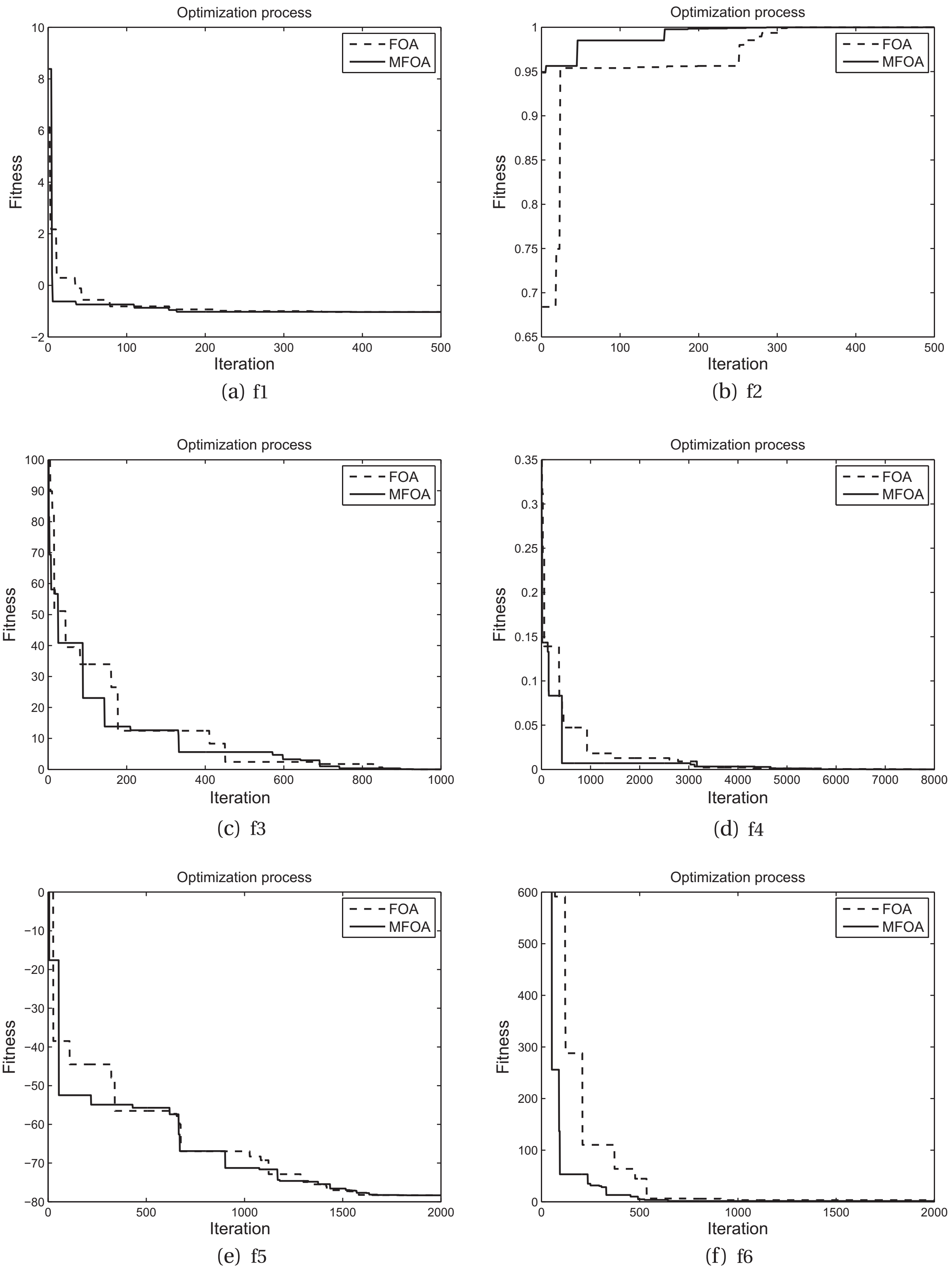

Fig. 6. Optimization process using FOA and MFOA approach for f1–f6.

Algorithm	Stats	f1 (-1.0316284)	f2 (1.0)	f3 (0.0)	f4 (0.0)	f5 (-78.332314)	f6 (0.0)
FOA	Best	-1.031609	0.999984	7.6e-006	3.3e-005	-78.332231	2.1e-004
	Worst	-1.029161	0.990281	0.994962	0.105843	-67.023039	0.497471
	Mean	-1.030697	0.998254	0.208622	0.041225	-75.852716	0.134255
	Rate	95%	85%	65%	70%	70%	60%
MFOA	Best	-1.031627	1.0	0.0	3.2e-006	-78.332302	2.5e-005
	Worst	-1.030214	0.997892	0.041249	0.022589	-72.678150	0.494948
	Mean	-1.031125	0.999163	0.015619	0.010638	-77.122705	0.113673
	Rate	100%	95%	85%	85%	80%	85%

Fig. 7. Simulation results of FOA and MFOA approach for f1–f6.

Algorithm	Stats	f1	f2	f3	f4	f5	f6
PSO	Mean	-1.031626	0.999984	0.011968	0.003795	-68.367517	1.798256
	Std	1.0e-06	7.0e-06	8.6e-03	3.0e-03	7.2e-00	8.8e-01
	Best	-1.031628	1.0	0.0	0.0	-78.332302	8.6e-004
CMAES	Mean	-1.022527	0.996273	0.070160	0.001008	-71.123059	0.254623
	Std	8.4e-03	1.8e-03	2.9e-02	8.9e-04	6.6e-00	1.0e-01
	Best	-1.031628	0.999925	0.029721	0.0	-78.332289	0.0
SaDE1	Mean	-1.031627	0.997836	0.045876	0.002766	-66.549080	0.387544
	Std	1.0e-06	1.7e-03	3.2e-02	2.0e-03	9.9e-00	3.1e-01
	Best	-1.031628	0.999998	0.000527	0.0	-78.332297	3e-006
SaDE2	Mean	-1.031628	0.999328	0.001673	0.007731	-72.207516	0.913510
	Std	0.0e-00	4.3e-04	7.3e-04	6.4e-03	5.8e-00	6.9e-01
	Best	-1.031628	1.0	0.000561	0.0	-78.332305	7.4e-004
MFOA	Mean	-1.031257	0.999097	0.015787	0.011274	-76.983460	0.120608
	Std	2.4e-04	7.4e-04	8.3e-03	5.2e-03	1.9e-00	4.8e-02
	Best	-1.031626	1.0	2e-06	3e-06	-78.332303	2.4e-05

Fig. 8. Simulation results of different algorithms for f1–f6.

Owing to stochastic nature, evolutionary algorithms may arrive at better or worse solutions than solutions they have previously reached during their search for new solutions. For this reason, it is beneficial to use statistical tools to compare the problem-solving success of one algorithm with that of another. The simple statistical parameters that can be derived from the results of an algorithm solving a specific numerical problem many times under different initial conditions – i.e. the mean solution (Mean), the standard deviation of the mean solution (Std) and the best solution (Best). The simulation results for f1–f6 with 30 runs using these techniques are reported in Fig. 8, which has also verified the effective performance of the proposed approach.

4.2. Parameter identification of synchronous generator

In this section, simulations are performed to evaluate the performance of the optimization algorithms for parameter identification of synchronous generator as in [29]. The mathematical model of synchronous generator is described as follows.

Electrical model of d-axis is described as:

$$\begin{aligned}
\frac{dE'_q}{dt} &= -\frac{1}{T'_{d0}}E'_q - \frac{1}{T'_{d0}}(X_d - X'_d)i_d + \frac{1}{T'_{d0}}\cdot\frac{X_{ad}}{R_{fd}}\cdot u_{fd} \\
\frac{dE''_q}{dt} &= \left(\frac{1}{T''_{d0}} - \frac{1}{T'_{d0}}\right)E'_q - \frac{1}{T''_{d0}}E''_q - \left(\frac{X_d - X'_d}{T'_{d0}} + \frac{X'_d - X''_d}{T''_{d0}}\right)i_d + \frac{1}{T'_{d0}}\cdot\frac{X_{ad}}{R_{fd}}\cdot u_{fd} \\
u_q &= E''_q - X''_d\cdot i_d
\end{aligned} \tag{21}$$

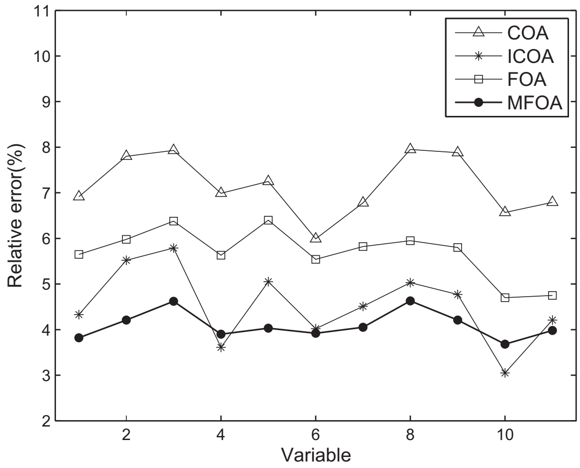

Fig. 9. Average relative error of each identification variable by different algorithms.

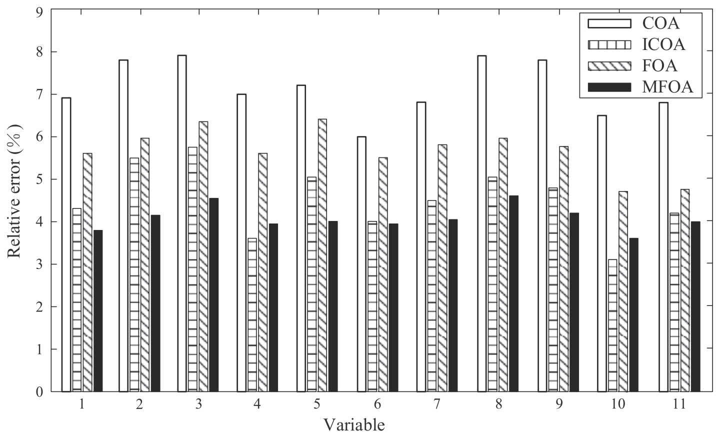

Fig. 10. Average relative error of each identification variable by different algorithms.

Electrical model of q-axis can be described as:

$$\begin{aligned} \frac{dE''_d}{dt} &= -\frac{1}{T''_{q0}}E''_d + \frac{X_q - X''_q}{T''_{q0}}i_q \\ u_d &= E''_d + X''_q \cdot i_q \end{aligned} \tag{22}$$

The equation of the motion is:

$$\begin{aligned} \frac{d\delta}{dt} &= \omega - \omega_0 \\ M\frac{d\omega}{dt} &= T_m - T_e - D(\omega - \omega_0) \end{aligned} \tag{23}$$

where subscript d and q denote the parameters of d- and q- axis, respectively. So there are 11 parameters values $(X_d, X'_d, X''_d, T'_{d0}, T''_{d0}, K, X_q, X''_q, T''_{q0}, M, D)$ to be identified, where $K = \frac{X_{ad}}{R_{fd}}$. The nominal values of synchronous generator are: rated power – 176.471 MVA, rated active power – 156.25 MW, rated voltage – 14.4 kV, power factor – 0.85, efficiency – 98.58%, rated speed – 3000 rpm, frequency – 50 Hz. The fitness function is: $min\ C(\hat{p}) = \sum_{k=1}^{n}[(i_d - \hat{i}_d)^2 + (i_q - \hat{i}_q)^2 + (\delta - \hat{\delta})^2]$, where the current i_d, i_q and the power angle δ are the measurable system outputs, the current $\hat{i}_d, \hat{i}_q$ and the power angle $\hat{\delta}$ are computed from the identified system model. These variables are calculated using standard per unit values, and the learning data-set with size of $n = 400$ are the same as in [29].

Here relative error is defined as: $\|\frac{x-\hat{x}}{x} * 100\%\|$, where x and $\hat{x}$ are the actual parameter value and the identified value, respectively. Average relative error of each variable by different algorithms repeated for 20 times have been shown in Figs.

9 and 10. The results of the chaos optimization algorithm (COA) [24], the improved chaos optimization algorithm (ICOA) in [29], original FOA, and the proposed MFOA are compared. In can seen from Figs. 9 and 10 that the MFOA approach has the smallest average relative error for this parameter identification problem, which has also verified the effective performance of the proposed approach.

5. Conclusion

The FOA is a new approach for finding global optimization based on the food finding behavior of the fruit fly. Multi-swarm FOA approach is proposed in this paper, where several sub-swarms moving independently in the search space with the aim of simultaneously exploring global optimal at the same time, and local behavior between sub-swarms are considered. Several simulations have shown the effective performance of the proposed approach. In addition, the parameters and sub-swarm number of the MFOA approach usually affects the search performance, and this may be improved in further study.

References

[1] H. Zang, S. Zhang, K. Hapeshi, A review of nature-inspired algorithms, J. Bionic Eng. 7 (S) (2010) S232–S237.
[2] E. Cuevas, M. Cienfuegos, D. Zaldivar, M. Perez-Cisneros, A swarm optimization algorithm inspired in the behavior of the social-spider, Expert Syst. Appl. 40 (16) (2013) 6374–6384.
[3] I.C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Inf. Process. Lett. 85 (6) (2003) 317–325.
[4] C.A.C. Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Trans. Evol. Comput. 8 (3) (2004) 256–279.
[5] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. Part B Cybern. 26 (1) (1996) 29–41.
[6] K.C. Ying, C.J. Liao, An ant colony system for permutation flow-shop sequencing, Comput. Oper. Res. 31 (5) (2004) 791–801.
[7] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (3) (2007) 459–471.
[8] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, Appl. Soft Comput. 8 (1) (2008) 687–697.
[9] W. Shen, X. Guo, C. Wu, D. Wu, Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm, Knowledge Based Syst. 24 (3) (2011) 378–385.
[10] X.S. Yang, S.S.S. Hosseini, A.H. Gandomi, Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect, Appl. Soft Comput. 12 (3) (2012) 1180–1186.
[11] F. Bartumeus, M.G.E. da Luz, G.M. Viswanathan, J. Catalan, Animal search strategies: a quantitative random walk analysis, Ecology 86 (11) (2005) 3078–3087.
[12] A.M. Reynolds, M.A. Frye, Free-flight odor tracking in Drosophila is consistent with an optimal intermittent scale-free search, PLOS ONE 2 (4) (2007) e354.
[13] Z.Z. Abidin, M.R. Arshad, U.K. Ngah, A simulation based fly optimization algorithm for swarms of mini autonomous surface vehicles application, Indian J. Geo-Mar. Sci. 40 (2) (2011) 250–266.
[14] W.T. Pan, A new fruit fly Optimization algorithm: taking the financial distress model as an example, Knowledge Based Syst. 26 (2012) 69–74.
[15] S.M. Lin, Analysis of service satisfaction in web auction logistics service using a combination of fruit fly optimization algorithm and general regression neural network, Neural Comput. Appl. 22 (3–4) (2013) 783–791.
[16] H. Li, S. Guo, C. Li, J. Sun, A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm, Knowledge Based Syst. 37 (2013) 378–387.
[17] P.W. Chen, W.Y. Lin, T.H. Huang, W.T. Pan, Using fruit fly optimization algorithm optimized grey model neural network to perform satisfaction analysis for e-business service, Appl. Math. Inf. Sci. 7 (2) (2013) 459–465.
[18] C. Li, S. Xu, W. Li, L. Hu, A novel modified fly optimization algorithm for designing the self-tuning proportional integral derivative controller, J. Convergence Inf. Technol. 7 (16) (2012) 69–77.
[19] Y.F. Xing, Design and optimization of key control characteristics based on improved fruit fly optimization algorithm, Kybernetes 42 (3) (2013) 466–481.
[20] W.T. Pan, Using modified fruit fly optimisation algorithm to perform the function test and case studies, Connection Sci. 25 (2–3) (2013) 151–160.
[21] L. Wang, X.L. Zheng, S.Y. Wang, A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem, Knowledge Based Syst. 48 (2013) 17–23.
[22] D. Shan, G.H. Cao, H.J. Dong, LGMS-FOA: an improved fruit fly optimization algorithm for solving optimization problems, Math. Prob. Eng. 37 (2013) 1–9.
[23] D.X. Yang, G. Li, G.D. Cheng, On the efficiency of chaos optimization algorithms for global optimization, Chaos Solitons Fractals 34 (4) (2007) 1366–1375.
[24] X.F. Yuan, Y.M. Yang, H. Wang, Improved parallel chaos optimization algorithm, Appl. Math. Comput. 219 (8) (2012) 3590–3599.
[25] R. Thangaraj, M. Pant, A. Abraham, P. Bouvry, Particle swarm optimization: hybridization perspectives and experimental illustrations, Appl. Math. Comput. 217 (12) (2011) 5208–5226.
[26] C. Igel, N. Hansen, S. Roth, Covariance matrix adaptation for multi-objective optimization, Evol. Comput. 15 (1) (2007) 1–28.
[27] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (6) (2006) 646–657.
[28] A.K. Qin, P.N. Suganthan, Self-adaptive differential evolution algorithm for numerical optimization, in: Proc. 2005 IEEE Cong. Evol. Comput. 2 (2005) 1785–1791.
[29] Q. Zhu, X.F. Yuan, H. Wang, An improved chaos optimization algorithm-based parameter identification of synchronous generator, Electr. Eng. 94 (3) (2012) 147–153.