Natural Computing (2020) 19:699–712
https://doi.org/10.1007/s11047-018-9712-z

Ensemble particle swarm optimization and differential evolution with alternative mutation method

H. Wang[1] · L. L. Zuo[1] · J. Liu[1] · W. J. Yi[1] · B. Niu[1]

Published online: 24 September 2018

Abstract
This paper presents a new ensemble algorithm which combines two well-known algorithms particle swarm optimization (PSO) and differential evolution (DE). To avoid the suboptimal solutions occurring in the previous hybrid algorithms, in this study, an alternative mutation method is developed and embedded in the proposed algorithm. The population of the proposed algorithm consists of two groups which employ two independent updating methods (i.e. velocity updating method from PSO and mutative method from DE). By comparing with the previously generated population at the last generation, two new groups are generated according to the updating methods. Based on the alternative mutation method, the population is updated by the alternative selection according to the evaluation functions. To enhance the diversity of the population, the strategies of re-mutation, crossover, and selection are conducted throughout the optimization process. Each individual conducts the correspondent mutation and crossover strategies according to the parameter values randomly selected, and the parameter values of scaling factor and crossover probability will be updated accordingly throughout the iterations. Numerous simulations on twenty-five benchmark functions have been conducted, which indicates the proposed algorithm outperforms some well-exploited algorithms (i.e. inertia weight PSO, comprehensive learning PSO, and DE) and recently proposed algorithms (i.e. DE with the ensemble of parameters and mutation strategies and ensemble PSO).

Keywords Particle swarm optimization · Differential evolution algorithm · Alternate mutation method · Ensemble strategy

1 Introduction

Particle swarm optimization (PSO) (Kennedy and Eberhart 1995) is known for its fast convergence, fewer initialization parameters, and easy to implement in complex optimization problems, which has been widely applied to many practical problems such as sampling-based image matting problem (Mohapatra et al. 2017), radial basis function networks problem (Alexandridis et al. 2016) and constrained non-convex and piecewise optimization problem (Chen et al. 2017). However, the main drawback associated with PSO and its variants is easily to fall into the local optima in comparison to other evolutionary algorithms. Different from the PSO method, differential evolution (DE) (Storn and Price 1997) is famous for its superior exploration capability using the strategies such as mutation, crossover and selection. Currently, DE has shown the great success in engineering applications such as economic or emission dispatch problem (Jebaraj et al. 2017), circuit designs problem (Zheng et al. 2017), and flood classification problem (Liao et al. 2013). Even so, the convergence speed of DE seems to be rather slow in the late optimization stage, which leads to the local optimum like PSO.

To overcome the limitation associated with PSO methods and DE methods, various strategies (Cheng et al. 2014; Guo et al. 2015; Juang et al. 2015; Niu et al. 2014, 2017) are used in the improvements of PSO and DE method. For example, some methods [e.g. integrating a unification factor (Tsai 2017), incorporating the individual particles memories (Guedria 2016) and improving the particles' collision and "territories" (Arani et al. 2013)] were applied in the independent PSO to avoid the premature convergence. Additionally, to encourage broader exploration, some strategies were also introduced in the independent DE, such as, the combination of Taguchi method with

✉ B. Niu
drniuben@gmail.com

1 College of Management, Shenzhen University, Shenzhen, China

sliding levels (Tsai 2015), the introduction of fuzzy selection method (Pandit et al. 2015) and the application of restricting the discrete variables (Ho-Huu et al. 2015). In recent years, the hybrid DE and PSO have been verified the superior performance on practical problems. In (Ma et al. 2015), a hierarchical hybrid algorithm, adopting the velocity and position method in PSO and a mutation strategy in DE, has been proposed and applied to solving the bi-level programming problem (BLPP). With equal sub lots method, a proposed hybrid algorithms of PSO and DE (Vijay Chakaravarthy et al. 2013) was used for scheduling m-machine flow shops with lot streaming. A hybrid method of PSO and DE integrating fuzzy c-means clustering algorithm had good performance on image segmentation (Liu and Qiao 2015).

Although the combination strategies can improve the effectiveness of global capability of DE or PSO, they generally have to burden larger computational complexity. To address this drawback, many improvements of hybrid PSO and DE have been proposed. In (Mao et al. 2017), combined the DE with the acceleration factors updating strategy in PSO, a global optimization method was developed to reduce the computational complexity. The updated mutation and crossover strategies were introduced in the novel hybrid DE and PSO algorithm (DE-PSO) (Xu et al. 2016) to make use of the shared resources, i.e. location and time. In (Tang et al. 2016), a novel hybrid PSO and DE algorithm, with the nonlinear time-varying PSO (NTVPSO) and the ranking-based self-adaptive DE (RBSADE), was developed to avoid stagnation and enhance the convergence speed. With the strategies of making uses of the population diversity of DE and the convergence ability of PSO, a multi-objective hybrid algorithm (Ma et al. 2015) integrating DE and PSO is designed to quickly produce the satisfactory solutions. In addition, some strategies have been presented in the hybrid DE and PSO methods to enhance the global search ability, such as the aging leader and challenger strategy (Moharam et al. 2016), population reduction strategy (Ali and Tawhid 2016), and hybrid operator and a multi-population strategy (Zuo and Xiao 2014). Though those hybrid methods can improve the performance of the original algorithms of the PSO and DE, premature stagnation is still a major problem.

In this study, based on an alternative mutation method, a combination of PSO and DE with ensemble strategies (EPSODE) is proposed to address the problem of premature convergence in original PSO and DE. Unlike previous combination algorithms of DE and PSO, this proposed EPSODE algorithm is a hierarchical method, which includes the alternative mutation method, the novel mutation and crossover strategies. Different mutation strategies of DE algorithm can achieve more accurate results than a unique mutation strategy. The ensemble mutation and crossover strategies have the characteristics of fast convergence and easy to jump out of the local optimal situation. To enhance population diversity, in the alternative mutation strategy, the population is separated into two groups generated by two different methods (i.e. velocity updating strategy of PSO and mutation strategy of DE). Additionally, those two new generated groups are updated by making the comparison with the previous population. The population at the next generation is obtained by intentionally selecting from two separated groups.

Thus, the main contributions of this paper are as follows:

- A new combination of PSO and DE is proposed with ensemble strategies to address the problem of premature convergence;
- The ensemble mutation and crossover strategies are developed to ensure the diversity of the population and the convergence speed of the optimization;
- The population of the proposed method is separately updated using velocity updating strategy of PSO and mutation strategy of DE to increase the disturbances between individuals;
- Compared with three well-exploited algorithms and two recently proposed PSO and DE algorithms, the new proposed combination algorithm is demonstrated to be superior by demonstrating on twenty-five benchmark functions.

The remaining paper is organized as follows. Section 2 provided a brief introduction of PSO algorithm and DE algorithm. In Sect. 3, the ensemble algorithm of PSO and DE with alternative mutation method is presented. Section 4 gives experimental results of the simulations on benchmark functions. Finally, Sect. 5 gives the conclusion remarks.

2 Description of algorithms

In this Section, the particle swarm optimization (PSO) method and differential evolution (DE) method are introduced, separately.

2.1 Particle swarm optimization

Inspired by the behavior of birds flocking, PSO (Kennedy and Eberhart 1995) is proposed as one of the intelligent optimization methods. In the simulation of PSO algorithm, a group of particles without quality and volume flies in the search space to find the optimal location. Each particle represents a potential solution to the problem under analysis. All particles are given random positions x and velocities v at the initial state. At each generation, the best position of ith particle is represented by $p\ best_i$ and the best position of all particles is denoted by $gbest$. By learning from them, the particles gradually approach to their optimal position. Equations (1)–(2) are the update equation:

$$v_i^{G+1} = \omega v_i^G + c_1 rand(0,1)(pbest_i^G - x_i^G) + c_2 rand(0,1)(gbest^G - x_i^G) \quad (1)$$

$$x_i^{G+1} = x_i^G + v_i^{G+1} \quad (2)$$

where N is the number of particles, D is the dimension of search space, and G is the number of generation, $x_i(i = 1, 2, \ldots, N)$ denotes the ith particle, $v_i(i = 1, 2, \ldots, N)$ is the velocity (i.e. the rate of change of position) of the ith particle. c_1 and c_2 are acceleration factors. Finally, the value of inertia weight represented by ω plays an important role in the search ability of PSO algorithm.

In addition to giving a fixed inertia weight, Shi and Eberhart (1998) proposed a linearly decreasing strategy for inertia weight [i.e. Eq. (3)] to improve the premature convergence in 1999. In the early part of the optimization, larger inertia weight can enhance the global exploration. While a smaller inertia weight in the later part of the generation can enable the local exploitation.

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) * G/G_{\max} \quad (3)$$

where ω_{max} and ω_{min} are the maximum and minimum values of inertia weight respectively. $G$is the current number of generation and G_{max} is the maximum number of generation.

2.2 Differential evolution algorithm

Proposed by Storn and Price (1997), DE is a well-known evolutionary optimization algorithm, which generally includes three basic operators including mutation, crossover, and selection. In DE algorithm, each candidate solution is encoded as $x_i = \{x_{i1}, x_{i2}, \ldots, x_{iD}\}$, i = 1, 2, …, N. The initial population should be distributed throughout the whole search space. The lower limit and the upper limit of the search space are $x_{\min} = \{x_{\min 1}, x_{\min 2}, \ldots, x_{\min D}\}$ and $x_{\max} = \{x_{\max 1}, x_{\max 2}, \ldots, x_{\max D}\}$, respectively. In general, the initial population is generated by using the following equation [i.e. Eq. (4)].

$$x_{ij} = x_{\min j} + rand(1,0) \cdot (x_{\max j} - x_{\min j}) \quad j = (1, 2, \ldots, D) \quad (4)$$

where N is the number of individuals and D is the search space's dimension.

Mutation: In order to avoid the local optima, the offspring of DE algorithm is produced by using mutation operation (Salman et al. 2007), and most frequently used mutation strategies are shown as follows:

$$\text{DE/rand/1 (Storn 1996)}: \quad V_i^G = x_{r1}^G + F \cdot (x_{r2}^G - x_{r3}^G) \quad (5)$$

$$\text{DE/rand/2 (Qin etal. 2009)}: \quad V_i^G = x_{r1}^G + F \cdot (x_{r2}^G - x_{r3}^G) + F \cdot (x_{r4}^G - x_{r5}^G) \quad (6)$$

$$\text{DE/best/1 (Storn 1996)}: \quad V_i^G = x_{best}^G + F \cdot (x_{r1}^G - x_{r2}^G) \quad (7)$$

$$\text{DE/best/2 (Storn 1996)}: \quad V_i^G = x_{best}^G + F \cdot (x_{r1}^G - x_{r2}^G) + F \cdot (x_{r3}^G - x_{r4}^G) \quad (8)$$

$$\text{DE/rand-to-best/1 (Storn 1996) or DE/target-to-best/1 (Price et al. 2005)}: \quad V_i^G = x_i^G + K \cdot (x_{best}^G - x_i^G) + F \cdot (x_{r1}^G - x_{r2}^G) \quad (9)$$

$$\text{DE/rand-to-best/2 (Qin et al. 2009)}: \quad V_i^G = x_i^G + K \cdot (x_{best}^G - x_i^G) + F \cdot (x_{r1}^G - x_{r2}^G + x_{r3}^G - x_{r4}^G) \quad (10)$$

$$\text{DE/current-to-rand/1 (Iorio and Li 2004)}: \quad u_i^G = x_i^G + K \cdot (x_{r1}^G - x_i^G) + F \cdot (x_{r2}^G - x_{r3}^G) \quad (11)$$

where G is the current generation, the scaling factor F plays an important role in disturbing the previous vectors. Larger scaling factor is useful for obtaining more potential solutions while smaller scaling factor facilitates to enhance the speed of convergence. K is a random number selected from 0 to 1. $r1$, $r2$, $r3$, $r4$ and $r5$ not equal to each other are randomly selected from 1 to N, and also different to i. V_i is the updated ith vector through mutating. x_{best}^G is the optimal individual at the Gth generation.

Though the single mutation strategy obtains good performance in DE, many limitations still exist. In terms of search accuracy, DE/rand/1 (Storn 1996) is widely used in the intelligence optimization field while it does not obtain the better solution relative to the DE/rand-to-best/1 (Storn 1996, Price et al. 2005). DE/rand/2 (Qin et al. 2009) increases the diversity relative to the DE/rand/1 (Storn 1996). DE/best/1 (Storn 1996, DE/rand-to-best/1 (Storn 1996, Price et al. 2005), DE/rand-to-best/2 (Qin et al. 2009) and DE/current-to-rand/1 (Iorio and Li 2004) all have certain limitations in high dimension and multi-model. Additionally, the speed of the DE/rand-to-best/1 (Storn 1996) is faster on easier optimization problems, and DE/current-to-rand/1 (Iorio and Li 2004) is superior to the other strategies for solving the rotated problems. Therefore, in order to integrate the advantages of those mutation strategies, the strategies of DE/rand-to-best/1 [i.e. Equation (9)] and DE/current-to-rand/1 [i.e. Eq. (11)] are employed in the proposed EPSODE algorithm.

Crossover: In DE algorithm, crossover operation is used for increasing the diversity of the population. The number of the alternative population is determined by the crossover probability *CR*. Smaller *CR* preserves the stability of the population during the evolution procedure, and larger *CR* enhances the diversity of the population. Equations (12)–(13) are the formulas of crossover:

$$u_{ij}^{G} = \begin{cases} V_{ij}^{G} & if(rand_j(0,1) \leq CR) or j = j_{rand} \\ x_{ij}^{G} & otherwise \end{cases}, j = 1 \ldots D \tag{12}$$

$$u_{ij}^{G} = \begin{cases} V_{ij}^{G} & ij = \langle j_{rand} \rangle_D, \langle j_{rand}+1 \rangle_D, \ldots, \langle j_{rand}+L-1 \rangle_D \\ x_{ij}^{G} & otherwise \end{cases} \tag{13}$$

where u_{ij} is the trial vector, j_{rand} is a randomly selected index in the range of[1, *D*], $rand_j$ is a random number in the range of [0, 1], $\langle l \rangle_D$ represents the modulo operation for *D*, and *L* is an integer between 1 to *D*. To enhance the diversity of the population, the binomial crossover and exponential crossover [i.e. Eq. (13)] (Zaharie 2009) are applied in the proposed algorithm (Mallipeddi and Suganthan 2010).

Selection: Following the crossover process, every trial vector$u_i$will be compared with the individual vector$x_i$in terms of the fitness value, and the vectors corresponding to the better fitness values will be preserved to the next generation. The greedy algorithm [i.e. Eq. (14)] is used to select individuals for the next generation process.

$$x_i^{G+1} = \begin{cases} u_i^G & if\, f(u_i^G) \leq f(x_i^G) \\ x_i^G & otherwise \end{cases} \tag{14}$$

3 Ensemble particle swarm optimization and differential evolution (EPSODE)

The superior characteristics of PSO are fast convergence speed and fewer initial parameters. However, due to insufficient information search, the suboptimal solutions might be more frequently obtained by PSO. Different from PSO, the population of DE tends to be more diverse as the number of generations increases but the computational complexity is greater. To take advantages of those two algorithms, in this paper, a new ensemble PSO and DE method (EPSODE) is proposed to improve the search capability of particles. The description of the proposed EPSODE algorithm is as follows.

3.1 Alternative mutation method

At the beginning of each generation, two subpopulation groups (i.e. P_1 and P_2) are generated by PSO and DE. Considering that PSO is easy to implement and has fewer parameters, one group (i.e., P_1) is produced by the PSO with the inertial weight of linearly decreasing strategy to reduce computational complexity [i.e. Eqs. (1)–(3)]. In order to disrupt the original movement direction, another group $P_2$is renewed by the mutation method [i.e. Eq. (15)].

$$x_i = K_1 \cdot (x_{pbest} - x_i) + K_2 \cdot (x_a - x_b),\ i = 1 \ldots N \tag{15}$$

where *N* is the individuals' number, K_1 and K_2 are randomly selected from [0, 1], $x_i$donates the *i*th individual. x_{pbest} is the optimal population, *a* and *b*are indexes selected from 1 to *N*, but different from the *i*.

The mutation method is proposed to break the rules of the original particle movement. For example, as shown in Fig. 1, if the initial directions of x_i, x_{pbest}, x_a and x_b are given, the directions of $x_{best} - x_i$ and $x_a - x_b$ are also determined. K_1 and K_2play the role in regulating the direction. If the value of $K_1$and K_2 are the same, the direction of the individual is the same of $(x_{pbest} - x_i) + (x_a - x_b)$. If not, it will be updated.

After updating the two subpopulation groups, the two updated groups (i.e. P_1 and P_2) are compared with the initial population or reserved population at the last generation according to the fitness value. The better individuals are preserved to update the new P_1andP_2again.

The individuals of two new groups (i.e. new P_1 and P_2) are sorted in accordance with the fitness value, and the sorted groups are also compared to retain the superior individuals. The new group P_3 is consist of the superior individuals, and served as the basic population in the next simulation experiment. The flowchart of the alternative mutation method is shown in Fig. 2.

3.2 Ensemble strategy of mutation and crossover

The selection method of mutation and crossover strategies plays an important role in the simulation process of DE. The suitable strategy can make the algorithm more efficient in solving the diffident types of functions, such as unimodal function, multimodal function, continuous functions, and discrete functions and so on. The recently proposed

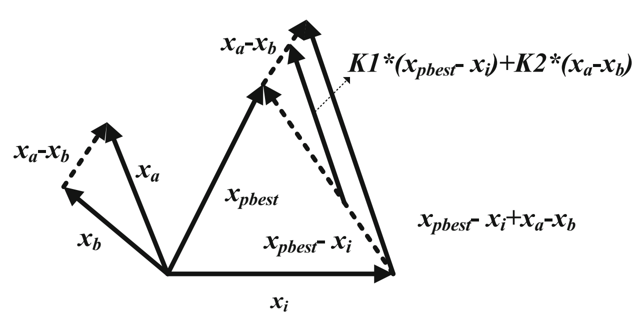

Fig. 1 Disturbance map of the improved mutation method

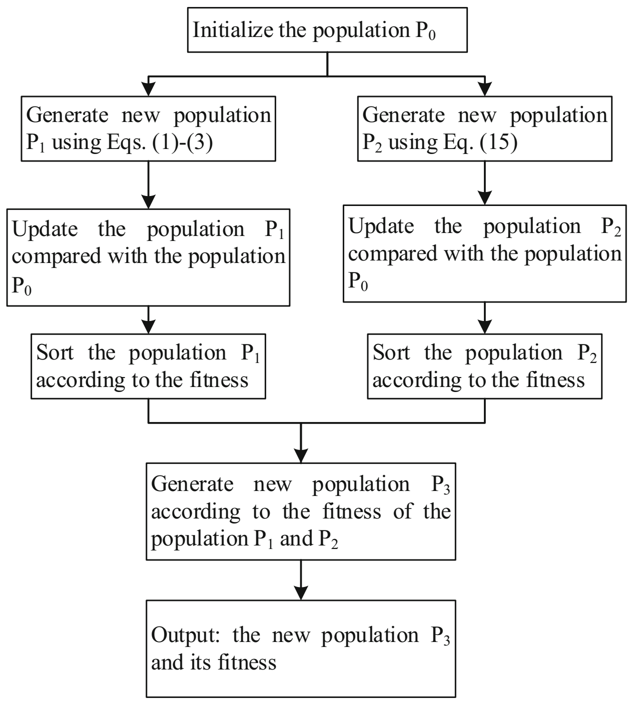

Fig. 2 The flowchart of the alternative mutation method

algorithm (Mallipeddi and Suganthan 2010) obtains superior offspring by employing the multi-strategies of mutation and crossover successfully. Based on the superior performance of multi-strategies, the ensemble of mutation and crossover strategy is employed in the proposed EPSODE algorithm.

In order to improve the global search ability of the proposed algorithm, DE/rand-to-best/1 [i.e. Eq. (9)] and DE/current-to-rand/1 [i.e. Eq. (11)] are applied in the proposed EPSODE algorithm. Getting information from the best individual of the whole group (i.e. DE/rand-to-best/1), the proposed EPSODE is easier to obtain the optimal value relative to other mutation strategies (i.e. DE/rand/1 and DE/rand/2). Obtaining information from neighbor individuals (i.e. DE/current-to-rand/1), the disturbance between individuals can be increased. Binomial crossover [i.e. Eq. (12)] and exponential crossover [i.e. (13)] (Wong et al. 2016) are applied in EPSODE to enhance population diversity.

In the ensemble method of mutation and crossover strategies, mutation crossover strategies are randomly selected according to the following pattern. The pattern is as follows: the scaling factor$F$is given two values (i.e. 0.5 and 0.9) and the crossover probability CR is also given three values (i.e. 0.1, 0.5, and 0.9). Each individual will be given a random$F$andCR during each iteration. T_1 and T_2 are the test labels, and the value is 0 or 1 (the two value can be selected arbitrarily). Each individual of the population $P3$ is given random T_1 and T_2. If $T_1 = 0$, the DE/rand-to-best/1

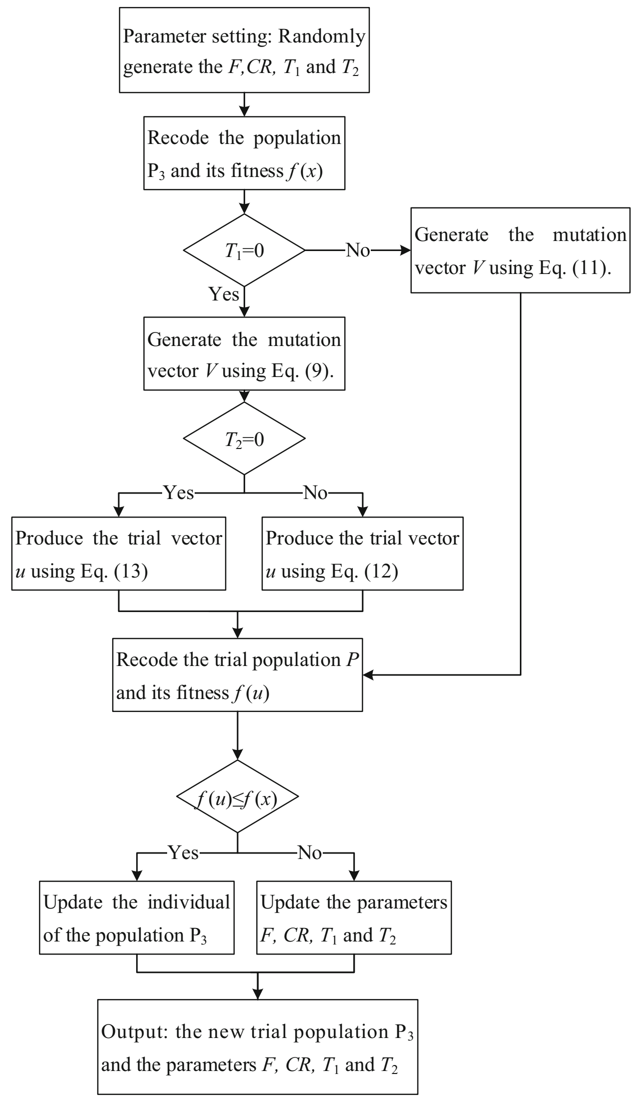

Fig. 3 The flowchart of the ensemble strategy of mutation and crossover

is selected as the mutation strategy according to the corresponding F. Otherwise, DE/current-to-rand/1 is used in EPSODE. To improve population diversity, the individuals which obtain information from the best individual of the whole group, are employed in the crossover strategy. If $T_1 = 0$ and $T_2 = 0$, the exponential crossover is selected as the crossover strategy according to the corresponding CR. If $T_1 = 0$ and $T_2 = 1$, binomial crossover is used for the crossover strategy. If the trial vector is not superior to the individual of populationP_3, the parameters (i.e. F, CR, T_1 and T_2) are updated again. The flowchart of the ensemble method of mutation and crossover strategies is shown in Fig. 3.

3.3 The pseudo-code of EPSODE

Effective termination criteria can save a lot of computation time while achieving superior solutions (Wong et al. 2016). In general, if the maximum number of generations is satisfied or the best value of fitness is found, the simulation process of this algorithm terminates. In EPSODE, the function evaluations have been performed more than once. Therefore, the original termination condition is prone to consume excessive computation because of the repeated fitness evaluation. To reduce computational complexity, the maximum number of function evaluations $FES$is employed as the termination criteria (Lynn and Suganthan 2017). The pseudo-code of EPSODE is given in Table 1.

As displayed in Table 1, in the simulation of EPSODE, the initial individual, generated using Eq. (4), is described as the initial target individual$x$. Based on the alternative mutation method, the population is updated using the updated velocity strategy of PSO and the modified mutation method of DE. The new trial individual u is generated according to the ensemble strategy of mutation and crossover. In order to obtained optimal generation, the function evaluation value of new trial individual$u$ is compared with the function evaluation value of target individual$x$. If the new trial individual$u$ is better than the target individual$x$, the trial individual$u$ is regarded as an updated target vector. Otherwise, the target vector $x$is reserved as the target individual of the next generation [i.e. Eq. (14)].

Table 1 The pseudo-code of EPSODE

The EPSODE algorithm
Parameters setting: c_1, c_2, ω_{max}, ω_{min}, F, CR, D, N, FES and G_{max}.
Initialization: $G=0$, $fitcount=0$, population P_0, velocity v_{id}.
F and CR is randomly selected by each individual.
Evaluate the fitness of the population P_0, and obtain $pbest$ and $gbest$.
$fitcount = fitcount + N$;
While $fitcount \le FES$
/*Alternative mutation method*/
Generate two new groups P_1 and P_2 using Eqs.(1)-(3) and Eq. (15).
$fitcount = fitcount + 2*N$;
Generate the new group P_3 according to the alternative mutation
method.
/*Mutation operation*/
For $i=1$ to N
Produce the new mutated vector V_i using the Eq. (9) or Eq. (11)
according to the corresponding F.
End
/*Crossover operation*/
For $i=1$ to N
For $j=1$ to D
Generate the trial vector u_i based on Eq. (12) or Eq. (13)
with random crossover probability CR.
End
End
/*Selection operation*/
For $i=1$ to N
Obtain the fitness value of each trial vector.
$fitcount = fitcount + 1$;
If $f(u_i) \le f(x_i)$
Then $x_i = u_i$; $pbest = x$;
If $f(x_i) \le f(gbest)$
Then $gbest = x_i$;
End
Else
F and CR is randomly selected by each individual again.
End
End
$G = G+1$;
If $G == G_{max}$
Then $G = G-1$;
End
End

4 Experiments and analyses

4.1 Benchmark functions and algorithms

Twenty-five benchmark functions (Suganthan et al. 2005) are applied to test the performance of the proposed EPSODE algorithm. According to the CEC2005 benchmark competition (Suganthan et al. 2005), those twenty-five functions can be divided into two categories: unimodal functions(i.e. functions $F1$–$F5$)and multimodal functions ($F6$–$F25$). Multimodal functions contain basic multimodal functions (i.e. functions $F6$–$F12$), expanded functions (i.e. functions $F13$–$F14$) and hybrid composition functions (i.e. functions $F15$–$F25$). The initialization ranges, search ranges and bias values of these benchmark functions are given in Table 2. The initialization range is set in accordance with CEC2005 (Suganthan et al. 2005).

To verify the performance of the proposed algorithm, some classic algorithms [i.e. PSO, DE and comprehensive learning PSO (CLPSO) (Liang et al. 2006)] and recently proposed algorithms [e.g. differential evolution algorithm with ensemble of parameters and mutation strategies (EPSDE) and ensemble PSO (EPSO) (Lynn and Suganthan 2017)] are introduced.

- Inertia weight PSO (PSO) (Shi and Eberhart 1998).
- Differential evolution algorithm (DE) (Storn and Price 1997).
- Differential evolution algorithm with ensemble of parameters and mutation strategies(EPSDE) (Mallipeddi and Suganthan 2010).
- Comprehensive learning PSO (CLPSO) (Liang et al. 2006).
- Ensemble particle swarm optimizer (EPSO) (Lynn and Suganthan 2017).

The first three algorithms are the original algorithms and have been used in the proposed EPSODE algorithm. In CLPSO algorithm, the historical best information of all

Table 2 CEC 2005 test functions

Functions	Initialization range	Search range	F(x*) *f_bias*
Unimodal functions			
F1: Shifted sphere function	[− 100,100]	[− 100,100]	− 450
F2: Shifted Schwefel's problem 1.2	[− 100,100]	[− 100,100]	− 450
F3: Shifted rotated high conditioned elliptic function	[− 100,100]	[− 100,100]	− 450
F4: Shifted Schwefel's problem 1.2 with noise in fitness	[− 100,100]	[− 100,100]	− 450
F5: Schwefel's problem 2.6 with global optimum on bounds	[− 100,100]	[− 100,100]	− 310
Multimodal functions			
F6: Shifted Rosenbrock's function	[− 100,100]	[− 100,100]	390
F7: Shifted rotated Griewank's function without bounds	[0,600]	[− 600,600]	− 180
F8: Shifted rotated Ackley's function with global optimum on bounds	[− 32,32]	[− 32,32]	− 140
F9: Shifted Rastrigin's function	[− 5,5]	[− 5,5]	− 330
F10: Shifted rotated Rastrigin's function	[− 5,5]	[− 5,5]	− 330
F11: Shifted rotated Weierstrass function	[− 0.5,0.5]	[− 0.5,0.5]	90
F12: Schwefel's problem 2.13	[− 100,100]	[− 100,100]	− 460
Expanded functions			
F13: Expanded extended Griewank's plus Rosenbrock's function (F8F2)	[− 3,1]	[− 3,1]	− 130
F14: Shifted rotated expanded Scaffer's F6	[− 100,100]	[− 100,100]	− 300
Hybrid composition functions			
F15: Hybrid composition function	[− 5,5]	[− 5,5]	120
F16: Rotated hybrid composition function	[− 5,5]	[− 5,5]	120
F17: Rotated hybrid composition function with noise in fitness	[− 5,5]	[− 5,5]	120
F18: Rotated hybrid composition function	[− 5,5]	[− 5,5]	10
F19: Rotated hybrid composition function with a narrow basin for the global optimum	[− 5,5]	[− 5,5]	10
F20: Rotated hybrid composition function with the global optimum on the bounds	[− 5,5]	[− 5,5]	10
F21: Rotated hybrid composition function	[− 5,5]	[− 5,5]	360
F22: Rotated hybrid composition function with High Condition Number Matrix	[− 5,5]	[− 5,5]	360
F23: Non-continuous rotated hybrid composition function	[− 5,5]	[− 5,5]	360
F24: Rotated hybrid composition function	[− 5,5]	[− 5,5]	260
F25: Rotated hybrid composition function without Bounds	[− 2,5]	[− 5,5]	260

other particles is applied to update the velocity. The CLPSO algorithm obtains good performances in dealing with the multimodal problem (Liang et al. 2006). EPSO algorithm, combined with a variety of PSO strategies, has been demonstrated to be superior in dealing with real-parameter optimization problems (Lynn and Suganthan 2017). All experiments are conducted through MATLAB R2014a software.

The same parameters of the six algorithms are described in detail (Lynn and Suganthan 2017). The maximum number of the generation ($G_{\max}$) is a constant value i.e. $G_{\max}$= 7500. The maximum number of function evaluations (*FES*) varies with the population size *N*and the maximum number of the generation $G_{\max}$, i.e. $FES{=}N * G_{\max}$ (Lynn and Suganthan 2017). Therefore, *FES*= 300,000 and *N* = 40 are used in the simulation of 30-dimensional problems. When the problem dimension is 50, *FES* is 600,000 and *N* is 80. If the maximum number of the generation $G_{\max}$ is satisfied while the maximum number of function evaluations (*FES*) is not satisfied, the current number of generation would decrease in order. Other parameters of all algorithms are shown as follow:

- In CLPSO (Liang et al. 2006), inertia weight$\omega$is also from 0.9 to 0.2, (i.e. ω_{max} = 0.9 and ω_{min} = 0.2), and acceleration coefficient c is from 3 to 1.5.
- In inertia weight PSO (Shi and Eberhart 1998), inertia weight$\omega$is from 0.9 to 0.2, $c_1 = 2$, $c_2 = 2$.
- In EPSO, some parameters of the integrated inertia weight PSO and CLPSO such as inertia weight ω and acceleration coefficient c, are same as above. Other parameters are from the literature (Lynn and Suganthan 2017).

Table 3 Experiment results on benchmark functions with dimensionality 30

	Result	PSO	EPSO	CLPSO	DE	EPSDE	EPSODE
F1	Mean	1.42E−13	5.68E−14	5.68E−14	*0.00E+00*	*0.00E+00*	*0.00E+00*
	SD	4.02E−14	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	Mean	2.41E−01	6.20E−11	1.89E+03	1.22E+03	*5.68E−14*	*5.68E−14*
	SD	2.52E−01	3.26E−11	5.66E+02	8.23E+02	0.00E+00	0.00E+00
F3	Mean	3.83E+06	2.34E+05	2.24E+07	1.20E+08	1.99E+05	*1.97E+05*
	SD	1.89E+06	2.29E+04	1.43E+07	6.98E+06	9.90E+03	7.33E+04
F4	Mean	3.84E+02	4.13E+02	1.25E+04	3.00E+03	2.63E−04	*1.02E−04*
	SD	3.69E+02	5.80E+01	1.79E+03	4.98E+02	3.71E−04	1.93E−05
F5	Mean	4.24E+03	4.93E+03	4.76E+03	2.55E+01	*1.70E+00*	1.77E+00
	SD	1.62E+01	2.30E+03	8.67E+02	2.56E+01	8.07E−01	2.50E+00
F6	Mean	8.44E+01	5.28E+00	9.77E+00	4.73E+01	2.84E−14	*1.99E−14*
	SD	8.85E+01	7.40E+00	1.27E+01	4.22E+01	4.02E−14	2.82E−14
F7	Mean	5.35E+03	*4.70E+03*	*4.70E+03*	*4.70E+03*	*4.70E+03*	*4.70E+03*
	SD	4.33E−01	0.00E+00	0.00E+00	9.09E−13	0.00E+00	0.00E+00
F8	Mean	2.10E+01	*2.09E+01*	2.10E+01	*2.09E+01*	*2.09E+01*	*2.09E+01*
	SD	4.00E−02	7.38E−02	1.40E−02	7.36E−02	1.86E−02	5.45E−02
F9	Mean	1.64E+01	2.49E+00	2.84E−14	8.54E+01	*0.00E+00*	*0.00E+00*
	SD	3.52E+00	2.11E+00	4.02E−14	7.05E+00	0.00E+00	0.00E+00
F10	Mean	1.46E+02	6.61E+01	1.00E+02	2.03E+02	*5.94E+01*	6.03E+01
	SD	6.32E+01	5.01E+00	5.96E−01	3.68E+00	1.57E+00	1.68E+01
F11	Mean	*2.08E+01*	2.81E+01	2.57E+01	3.97E+01	3.12E+01	2.94E+01
	SD	2.29E+00	1.08E+00	1.70E+00	1.50E−01	1.43E+00	8.00E−01
F12	Mean	6.70E+04	*1.71E+04*	1.96E+04	4.20E+04	1.90E+04	2.36E+04
	SD	3.63E+04	1.61E+04	8.56E+03	3.05E+04	1.59E+04	9.57E+03
F13	Mean	2.93E+00	1.91E+00	1.94E+00	1.48E+01	2.14E+00	*1.89E+00*
	SD	7.85E−01	4.57E−02	5.28E−01	4.42E−02	2.57E−01	1.37E−01
F14	Mean	*1.23E+01*	1.28E+01	1.27E+01	1.36E+01	1.30E+01	1.29E+01
	SD	3.01E−01	6.28E−01	4.94E−01	1.05E−01	2.61E−02	2.76E−01
F15	Mean	3.51E+02	*2.02E+02*	5.51E+01	*2.00E+02*	*2.00E+02*	*2.00E+02*
	SD	2.14E+02	2.79E+00	1.55E+01	0.00E+00	0.00E+00	0.00E+00
F16	Mean	1.07E+02	1.11E+02	2.02E+02	2.95E+02	1.26E+02	*8.20E+01*
	SD	4.06E+01	4.61E+01	2.13E+01	2.64E+00	4.60E+01	8.53E+00
F17	Mean	3.75E+02	*8.04E+01*	2.25E+02	2.46E+02	1.45E+02	1.50E+02
	SD	2.33E+02	7.92E+00	1.60E+01	3.67E+00	5.15E+00	3.07E+01
F18	Mean	9.25E+02	9.06E+02	9.09E+02	9.06E+02	9.06E+02	*9.05E+02*
	SD	7.55E−01	2.74E−03	1.57E+00	9.81E−02	2.45E+00	2.00E+00
F19	Mean	9.29E+02	9.09E+02	9.08E+02	9.06E+02	*9.04E+02*	*9.04E+02*
	SD	2.54E+00	6.05E−01	4.28E−01	4.87E−02	7.69E−01	3.44E−01
F20	Mean	9.25E+02	9.09E+02	9.09E+02	9.05E+02	9.04E+02	*9.01E+02*
	SD	6.64E−01	8.41E−01	5.69E−01	1.24E−02	3.18E−01	2.97E+00
F21	Mean	*5.00E+02*	*5.00E+02*	6.18E+02	7.93E+02	*5.00E+02*	*5.00E+02*
	SD	0.00E+00	0.00E+00	2.10E+01	4.14E+02	3.22E−13	3.22E−13
F22	Mean	9.37E+02	*8.46E+02*	9.17E+02	8.63E+02	8.58E+02	8.60E+02
	SD	1.76E+01	9.23E+00	9.29E−01	1.09E+00	3.54E+00	4.36E+00
F23	Mean	*5.34E+02*	8.20E+02	6.33E+02	8.25E+02	*5.34E+02*	*5.34E+02*
	SD	6.26E−04	4.04E+02	3.01E+01	3.83E+02	1.61E−13	2.52E−04
F24	Mean	3.50E+02	9.39E+02	9.57E+02	9.49E+02	*2.00E+02*	*2.00E+02*
	SD	2.12E+02	1.79E+00	4.09E+00	1.41E+00	0.00E+00	0.00E+00
F25	Mean	1.12E+03	9.81E+02	9.97E+02	9.88E+02	*9.79E+02*	9.81E+02
	SD	1.36E+01	7.95E+00	2.24E+00	1.82E+00	5.58E+00	3.12E+00

Table 4 Experiment results on benchmark functions with dimensionality 50

	Result	PSO	EPSO	CLPSO	DE	EPSDE	EPSODE
F1	Mean	1.99E−13	8.53E−14	1.14E−13	5.68E−14	*0.00E+00*	*0.00E+00*
	SD	4.02E−14	4.02E−14	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	Mean	9.14E+02	*2.29E−07*	2.56E+04	8.82E+04	2.58E−03	2.24E−03
	SD	2.15E+02	1.35E−07	6.37E+03	3.97E+03	7.94E−04	2.22E−03
F3	Mean	2.65E+07	1.09E+06	7.81E+07	6.80E+08	4.63E+06	*5.77E+05*
	SD	3.35E+07	6.54E+05	1.49E+07	7.33E+07	1.80E+06	1.01E+06
F4	Mean	1.17E+04	3.80E+03	5.92E+04	1.03E+05	6.95E+01	*4.00E+01*
	SD	8.47E+02	1.26E+03	2.30E+03	6.50E+03	2.45E+01	6.94E+00
F5	Mean	7.95E+03	9.90E+03	1.29E+04	5.45E+03	*1.58E+03*	1.65E+03
	SD	1.58E+02	3.10E+03	1.14E+03	3.03E+02	5.18E+02	5.22E+02
F6	Mean	6.35E+01	2.35E+01	6.02E+00	4.29E+01	*1.33E−07*	1.99E+00
	SD	3.78E+01	4.06E+00	1.72E+00	1.09E+00	1.75E−07	2.82E+00
F7	Mean	7.16E+03	*6.20E+03*	*6.20E+03*	*6.20E+03*	*6.20E+03*	*6.20E+03*
	SD	4.87E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	9.09E−13
F8	Mean	2.12E+01	*2.10E+01*	2.11E+01	2.11E+01	2.11E+01	2.11E+01
	SD	1.02E−02	1.42E−02	4.25E−02	1.75E−02	6.64E−02	2.87E−02
F9	Mean	3.33E+01	8.57E+00	4.97E−01	2.29E+02	2.84E−14	*1.42E−14*
	SD	6.33E+00	1.95E+00	7.04E−01	6.42E+00	4.02E−14	4.02E−14
F10	Mean	1.94E+02	1.13E+02	2.96E+02	3.96E+02	1.63E+02	*9.76E+01*
	SD	1.87E+02	1.27E+01	8.22E+00	7.66E+00	1.54E+01	3.75E+00
F11	Mean	*4.73E+01*	*4.73E+01*	5.05E+01	7.26E+01	5.67E+01	5.64E+01
	SD	1.41E+00	3.22E+00	1.28E+00	7.95E−01	2.23E+00	9.87E−01
F12	Mean	3.68E+05	2.16E+05	9.05E+04	1.08E+06	1.50E+05	*7.58E+04*
	SD	2.65E+05	6.29E+04	4.77E+03	1.03E+05	3.01E+03	2.49E+04
F13	Mean	5.57E+00	4.06E+00	3.92E+00	3.04E+01	4.96E+00	*2.32E+00*
	SD	5.78E−01	2.08E−01	7.77E−02	2.63E+00	2.02E−02	6.76E−01
F14	Mean	2.25E+01	2.27E+01	*2.22E+01*	2.31E+01	2.27E+01	2.26E+01
	SD	1.06E−01	2.03E−02	6.78E−01	4.97E−01	2.16E−01	3.74E−01
F15	Mean	4.01E+02	2.57E+02	7.79E+01	*2.00E+02*	*2.00E+02*	3.00E+02
	SD	7.22E−01	6.06E+01	5.44E+00	2.33E−12	0.00E+00	0.00E+00
F16	Mean	1.02E+02	8.58E+01	2.32E+02	3.03E+02	1.32E+02	*6.38E+01*
	SD	1.93E+01	9.29E+00	2.65E+01	2.29E+00	8.32E−01	4.87E+00
F17	Mean	3.39E+02	9.36E+01	3.58E+02	3.28E+02	2.16E+02	*1.02E+02*
	SD	1.31E+02	3.76E+00	1.99E+01	1.32E+01	6.59E+00	1.71E+01
F18	Mean	9.56E+02	9.50E+02	9.27E+02	9.17E+02	9.15E+02	*9.05E+02*
	SD	1.45E−01	4.94E+00	3.85E+00	8.83E−01	6.61E−01	3.84E−01
F19	Mean	9.49E+02	9.41E+02	9.26E+02	9.17E+02	9.17E+02	*9.05E+02*
	SD	6.23E−01	5.23E+00	1.34E+00	9.66E−01	5.16E−01	5.19E−01
F20	Mean	9.55E+02	9.21E+02	9.29E+02	9.16E+02	9.16E+02	*9.06E+02*
	SD	9.82E+00	3.26E+00	2.71E+00	7.44E−03	6.31E−01	8.55E−01
F21	Mean	*9.39E+02*	1.01E+03	1.02E+03	1.01E+03	1.01E+03	1.01E+03
	SD	1.96E+02	2.39E+00	1.62E+00	4.78E+00	1.60E−01	3.41E+00
F22	Mean	9.79E+02	9.32E+02	9.43E+02	9.10E+02	8.99E+02	*8.88E+02*
	SD	5.16E+00	3.13E+01	7.91E+00	6.71E−01	3.35E+00	1.61E+00
F23	Mean	1.07E+03	7.80E+02	1.02E+03	1.01E+03	1.01E+03	*9.92E+02*
	SD	7.09E+00	3.40E+02	2.83E−01	1.79E−01	3.56E−01	1.30E+00
F24	Mean	*2.00E+02*	9.75E+02	1.02E+03	9.84E+02	1.00E+03	*2.00E+02*
	SD	0.00E+00	4.09E+00	8.71E+00	2.46E+00	4.24E+00	0.00E+00
F25	Mean	1.27E+03	1.23E+03	1.26E+03	*1.16E+03*	1.19E+03	1.20E+03
	SD	4.60E+00	6.14E+00	3.14E+00	9.67E+00	1.43E+01	1.27E+01

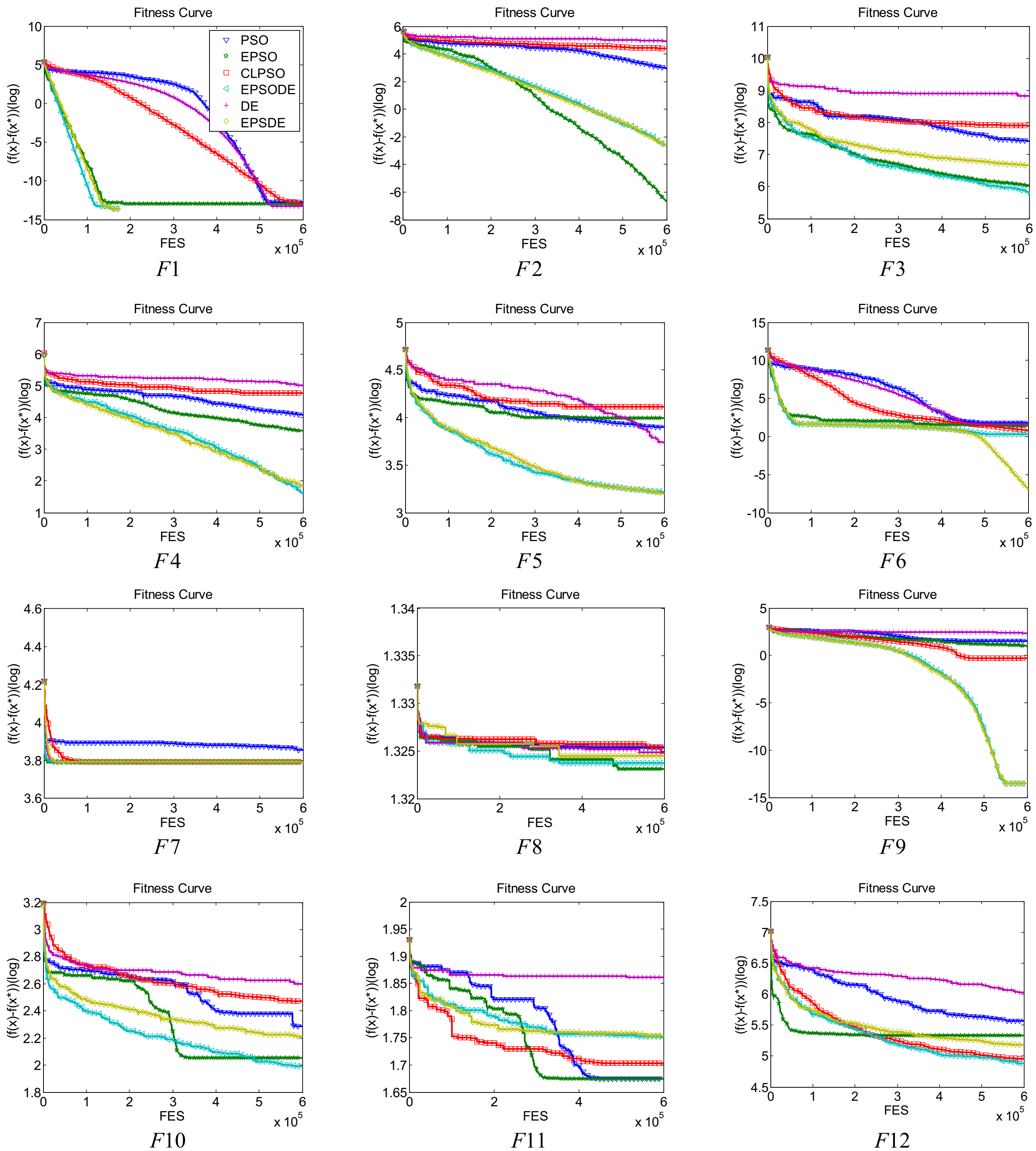

Fig. 4 The generation process of the algorithms when dimensionality = 50

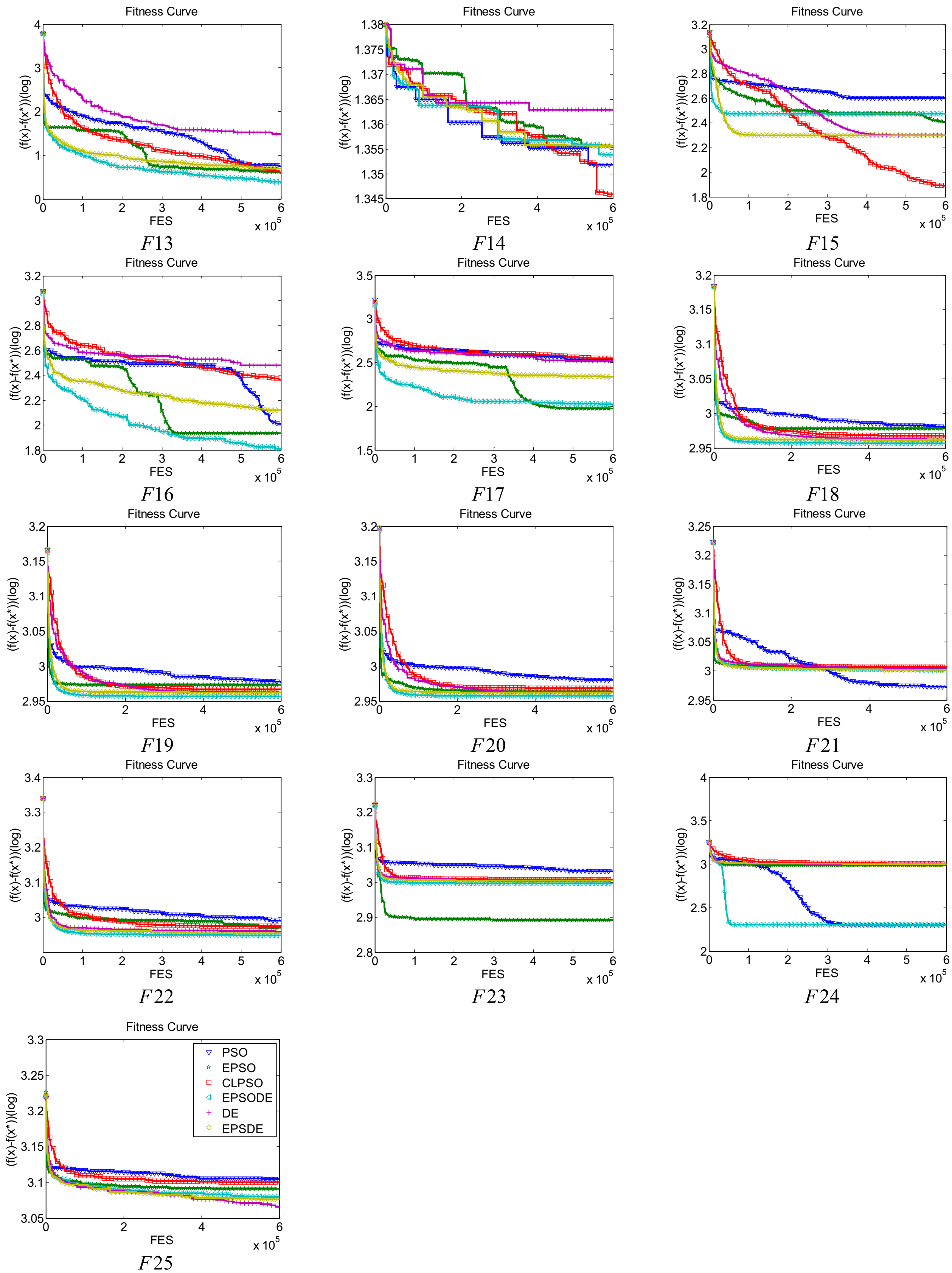

Fig. 4 continued

- In DE (Storn and Price 1997), mutation factor$F$is 0.9, and crossover probability CR is 0.5.
- In EPSDE (Mallipeddi and Suganthan 2010), mutation factors F are 0.5 and 0.9, and crossover probabilities CR are 0.1, 0.5 and 0.9.
- In the EPSODE, mutation factors F are also 0.5 and 0.9, and crossover probabilities CR are 0.1, 0.5 and 0.9. Inertia weight ω is also from 0.9 to 0.2, $c_1 = 2, c_2 = 2$.

4.2 Experiment results and discussion

The mean fitness and the standard deviation obtained by six algorithms are given in Tables 3 and 4. The best results among these optimization algorithms are highlighted in italics. Figure 4 shows the convergence curves of different test functions with the dimension 50. In order to make the graphical curve clear, the graphical interpretation labels are provided only in the first and last functions (i.e.$F1$ and $F25$). The legends in remaining subfigures are the same to $F1$ and $F25$.

The maximum number of function evaluations (*FES*) 300,000 and population size 40 are used in the simulation of 30-dimensional problems (Lynn and Suganthan 2017). Experiment results are illustrated in Table 3. For unimodal functions, EPSODE obtains the best results on functions $F1$–$F4$, and the second-best result on the function $F5$. EPSODE performs well as the EPSDE on functions $F1$ and $F2$. For basic multimodal functions, EPSODE performs best on functions $F6$–$F9$. EPSDE algorithm yields the best results on function$F10$. For the remaining basic multimodal functions (i.e. functions $F11$ and $F12$), inertial weight PSO and EPSO respectively get the best results. The proposed EPSODE algorithm dedicates the best results on the function $F13$ while inertial weight PSO complies the best on function$F14$. For hybrid composition functions, the proposed EPSODE algorithm implements the superior results on functions $F15$–$F16$, $F18$–$F21$ and $F23$–$F24$. EPSO algorithm performs the best on $F17$ and $F22$. EPSODE algorithm and EPSDE algorithm obtain the same fitness values on functions $F19$, $F21$ and $F23$, but the standard deviations of EPSODE are better than EPSDE except the function $F23$. Therefore, the proposed algorithm obtains good performance in most functions whether the function is unimodal or multimodal.

The experiment results in 50-dimensional problems are shown in Table 4. The number of function evaluations (FES) is 600,000 and population size is 80. EPSODE implements the best results on the functions $F1$ and $F3$–$F4$ while EPSO and EPSDE respectively perform the best on functions $F2$ and $F5$. For basic multimodal functions, EPSODE, EPSO, CLPSO, DE and EPSDE obtain the same best results on functions $F7$. The performances of EPSODE algorithm are significantly better than other algorithms on function $F9$–$F10$ and $F12$. For expanded functions (i.e. functions $F13$–$F14$), the proposed EPSDE successes to maintain its good performance on functions $F13$. For hybrid composition functions, DE and EPSDE comply the same best results on function $F15$. EPSODE obtains the best results on functions $F16$–$F20$ and $F22$–$F23$. Inertia weight PSO algorithm gets the best results on function $F21$. The proposed EPSODE gets the same performances as well as inertial weight PSO on function $F24$, but the convergence speed of EPSODE is evidently superior to the inertial weight PSO in Fig. 4. Therefore, EPSODE algorithm successful remained its superior performance in higher dimension problems.

As shown in Fig. 4, the convergence speed of EPSODE is faster than other algorithms on functions $F1$, $F7$, $F10$ and $F24$. The convergence solutions are closer to optimal values on most functions e.g., $F1$, $F3$, $F10$, $F13$, $F16$, $F24$ and so on. Altogether, the superior performance of EPSODE algorithm can still be seen from the convergence curves.

In summary, the proposed algorithm is more powerful than the other algorithms (i.e. inertial weight PSO, CLPSO, EPSO, DE and EPSDE) in terms of the search ability. Though EPSDE, and inertial weight PSO algorithm can obtain the similar solutions on some benchmark functions (e.g., functions $F1$ and $F24$), the convergence speed is not better than the proposed EPSODE algorithms. In terms of dimensionality, the proposed EPSODE algorithm performs better than other algorithms when the dimension increases. The main reasons of the superiority are as follows:

- One group uses the updating method of PSO to carry out more in-depth exploration in the alternative mutation method of EPSODE.
- The other group applies the mutative method of DE to disturb the original direction in the alternative mutation method of EPSODE.
- Hence, the new population is given a deeper ability to explore and is different from the original updating direction.
- Additionally, the non-single and fixed mutation crossover strategy contributes to increasing the diversity of the population.

5 Conclusion

An ensemble PSO and DE algorithm (EPSODE), based on the alternative mutation method, is proposed for solving different types of functions. Modified DE algorithm are the main program and PSO algorithm is the subprogram in the proposed algorithm. The velocity update strategy of PSO combined with the modified mutation method is the key of the proposed algorithm, which can avoid the premature convergence and improve the search capability. Meanwhile, the strategies of multiple mutation and crossover can improve convergence speed. Experiment results show that EPSODE algorithm outperforms other algorithms in terms of mean and standard deviation. Thus, the proposed alternative mutation method can enhance the performance of EPSODE algorithm, which has been verified by testing on the benchmark functions. In our future studies, the proposed algorithm will be developed and applied to solving the real-world problems.

Acknowledgements This work is partially supported by The National Natural Science Foundation of China (Grants Nos. 71571120, 71271140, 71471158, 71001072, 61472257), Natural Science Foundation of Guangdong Province (2016A030310074), Shenzhen Science and Technology Plan (CXZZ20140418182638764), and Research Foundation of Shenzhen University (85303/00000155).

References

Alexandridis A, Chondrodima E, Sarimveis H (2016) Cooperative learning for radial basis function networks using particle swarm optimization. Appl Soft Comput 49:485–497

Ali AF, Tawhid MA (2016) A hybrid PSO and DE algorithm for solving engineering optimization problems. Appl Math Inf Sci 10:431–449

Arani BO, Mirzabeygi P, Panahi MS (2013) An improved PSO algorithm with a territorial diversity-preserving scheme and enhanced exploration–exploitation balance. Swarm Evolut Comput 11:1–15

Chen JJ, Zheng JH, Wu P, Zhang LL, Wu QH (2017) Dynamic particle swarm optimizer with escaping prey for solving constrained non-convex and piecewise optimization problems. Expert Syst Appl 86:208–223

Cheng MY, Tran DH, Wu YW (2014) Using a fuzzy clustering chaotic-based differential evolution with serial method to solve resource-constrained project scheduling problems. Autom Constr 37:88–97

Guedria NB (2016) Improved accelerated PSO algorithm for mechanical engineering optimization problems. Appl Soft Comput 40:455–467

Guo SM, Yang CC, Hsu PH, Tsai SH (2015) Improving differential evolution with a successful-parent-selecting framework. IEEE Trans Evol Comput 19:717–730

Ho-Huu V, Nguyen-Thoi T, Nguyen-Thoi MH, Le-Anh L (2015) An improved constrained differential evolution using discrete variables (D-ICDE) for layout optimization of truss structures. Expert Syst Appl 42:7057–7069

Iorio AW, Li X (2004) Solving rotated multi-objective optimization problems using differential evolution. Lect Notes Comput Sci Inf Syst 3339:861–872

Jebaraj L, Venkatesan C, Soubache I, Rajan CCA (2017) Application of differential evolution algorithm in static and dynamic economic or emission dispatch problem: a review. Renew Sustain Energy Rev 77:1206–1220

Juang CF, Chen YH, Jhan YH (2015) Wall-following control of a hexapod robot using a data-driven fuzzy controller learned through differential evolution. IEEE Trans Ind Electron 62:611–619

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: IEEE international conference on neural networks, 1995. Proceedings. pp 1942–1948 vol. 1944

Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans Evol Comput 10:281–295

Liao L, Zhou J, Zou Q (2013) Weighted fuzzy kernel-clustering algorithm with adaptive differential evolution and its application on flood classification. Nat Hazards 69:279–293

Liu J, Qiao S (2015) A image segmentation algorithm based on differential evolution particle swarm optimization fuzzy c-means clustering. Comput Sci Inf Syst 12:873–893

Lynn N, Suganthan PN (2017) Ensemble particle swarm optimizer. Appl Soft Comput 55:533–548

Ma W, Wang M, Zhu X (2015) Hybrid particle swarm optimization and differential evolution algorithm for bi-level programming problem and its application to pricing and lot-sizing decisions. J Intell Manuf 26:471–483

Mallipeddi R, Suganthan PN (2010) Differential evolution algorithm with ensemble of parameters and mutation and crossover strategies. In: International conference on swarm, evolutionary, and memetic computing, pp 71–78

Mao B, Xie Z, Wang Y, Handroos H, Wu H, Shi S (2017) A hybrid differential evolution and particle swarm optimization algorithm for numerical kinematics solution of remote maintenance manipulators. Fusion Eng Des 124:587–590

Mohapatra P, Das KN, Roy S (2017) A modified competitive swarm optimizer for large scale optimization problems. Appl Soft Comput 59:340–362

Moharam A, El-Hosseini MA, Ali HA (2016) Design of optimal PID controller using hybrid differential evolution and particle swarm optimization with an aging leader and challengers. Appl Soft Comput 38:727–737

Niu B, Zhang F, Li L, Wu L (2014) Particle swarm optimization for yard truck scheduling in container terminal with a cooperative strategy. In: International conference on information science, electronics and electrical engineering, pp 1392–1396

Niu B, Huang HL, Tan LJ, Duan QQ (2017) Symbiosis-based alternative learning multi-swarm particle swarm optimization. IEEE/ACM Trans Comput Biol Bioinform 14:4–14

Pandit M, Srivastava L, Sharma M (2015) Environmental economic dispatch in multi-area power system employing improved differential evolution with fuzzy selection. Appl Soft Comput 28:498–510

Price K, Storn RM, Lampinen JA (2005) Differential evolution: a practical approach to global optimization (Natural Computing Series). vol 2. Springer, New York, Inc. Secaucus, NJ, USA ©2005

Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Comput 13:398–417

Salman A, Engelbrecht AP, Omran MGH (2007) Empirical analysis of self-adaptive differential evolution. Eur J Oper Res 183:785–804

Shi Y, Eberhart R (1998) A modified particle swarm optimizer. Springer, Berlin, Advances in Natural Computation

Storn R (1996) On the usage of differential evolution for function optimization. In: Fuzzy information processing society, 1996. NAFIPS. 1996 Biennial Conference of the North American, pp 519–523

Storn R, Price K (1997) Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359

Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y-P, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. IEEE congress on evolutionary computation

Tang B, Zhu Z, Luo J (2016) Hybridizing particle swarm optimization and differential evolution for the mobile robot global path planning. Int J Adv Rob Syst 13(3):1

Tsai JT (2015) Improved differential evolution algorithm for nonlinear programming and engineering design problems. Neurocomputing 148:628–640

Tsai HC (2017) Unified particle swarm delivers high efficiency to particle swarm optimization. Appl Soft Comput 55:371–383

Vijay Chakaravarthy G, Marimuthu S, Naveen Sait A (2013) Performance evaluation of proposed differential evolution and particle swarm optimization algorithms for scheduling m-machine flow shops with lot streaming. J Intell Manuf 24:175–191

Wong JYQ, Sharma S, Rangaiah GP (2016) Design of shell-and-tube heat exchangers for multiple objectives using elitist non-dominated sorting genetic algorithm with termination criteria. Appl Therm Eng 93:888–899

Xu J, Tang Y, Liu DY (2016) Research of hybrid differential evolution and particle swarm optimization algorithm using map reduce to schedule tasks. J Chin Comput Syst 37:1479–1481

Zaharie D (2009) Influence of crossover on the behavior of differential evolution algorithms. Appl Soft Comput 9:1126–1138

Zheng LM, Zhang SX, Zheng SY, Pan YM (2017) Differential evolution algorithm with two-step subpopulation strategy and its application in microwave circuit designs. IEEE Trans Ind Inf 12:911–923

Zuo X, Xiao L (2014) A DE and PSO based hybrid algorithm for dynamic optimization problems. Soft Comput 18:1405–1424