Expert Systems With Applications 200 (2022) 117118

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Poplar optimization algorithm: A new meta-heuristic optimization technique for numerical optimization and image segmentation

Debao Chen [a], Yuanyuan Ge [a], Yujie Wan [a], Yu Deng [b], Yuan Chen [b], Feng Zou [b,*]

[a] School of Computer Science and Technology, Huaibei Normal University, Huaibei 235000, China
[b] School of Physics and Electronic Information, Huaibei Normal University, Huaibei 235000, China

ARTICLE INFO

Keywords:
Poplar optimization algorithm
Sexual propagation
Asexual reproduction
Optimization methods
Benchmarks functions
Image segmentation

ABSTRACT

A novel algorithm called Poplar Optimization Algorithm (POA) is developed in this paper to solve continuous optimization problems. The algorithm mimics the sexual and asexual propagation mechanism of poplar, where the basic philosophy of how to execute sexual and asexual propagation for individuals is detail designed in the algorithm. Mutation strategy of backtracking search algorithm is adopted in POA to maintain the diversity in a certain degree. The performance of POA algorithm is tested on 25 functions from the CEC2005 test suite and 30 functions from the CEC2017 test suite with different features. The results of POA are compared with some other population-based algorithms in terms of the quality and efficiency. Finally, the proposed algorithm is used to find the optimal threshold for image segmentation. The results indicate that the poplar optimization algorithm can obtain competitive or superior performance.

1. Introduction

In real engineering fields, optimization problems are widely existed. Many difficulties such as parameter optimization, structure optimization and path optimization are all associated with the complex optimization. To solve these optimization problems, some optimization methods such as linear programming, dynamic programming, steepest decent, etc. are developed in last few decades. As gradient information is a necessary required condition for most of the traditional optimization methods, it is impossible to handle non-differentiable functions optimization problems. Moreover, the traditional methods are difficult to handle - multimodal problems, while local convergence is often appeared when there are many local optima in the optimization problems. Finding or designing some appropriate optimization methods for complex optimization problems is a hot point in intelligence optimization field. Evolutionary computing methods and swarm intelligence algorithms are proved to perform better than the traditional optimization methods and they are widely utilized in many recent optimization fields.

Various evolutionary computing methods and swarm intelligence algorithms are developed for handling complex optimization problems in the past few decades. These meta-heuristics don't need the gradient information or the accurate mathematical model, thus being widely utilized to handle non-differentiable optimization problems. Genetic algorithm (GA) (Holland, 1975) mimics the principle of survival of fittest of the Darwinian, uses selection, crossover, and mutation as the basic operations. GA and some improved GAs (Kumar, Husain, Upreti & Gupta, 2010) are widely used in different engineering and science fields. To mimic foraging behavior of different animals, particle swarm optimization (PSO) (Kennedy & Eberhart, 1995), ant colony optimization (ACO) (Dorigo, 1992) and artificial bee colony (ABC) (Akay & Karaboga, 2012) are developed for solving optimization problems. These algorithms are generally called swarm intelligence (SI) optimization algorithms. Among these three algorithms, PSO is a very active algorithm in swarm intelligence (SI) optimization and some improved variants are developed in recent years. A new learning strategy is developed in comprehensive learning particle swarm optimization algorithm (CLPSO) (Liang, Qin, Suganthan & Baskar, 2006) to decrease the premature convergence probability. The effectiveness of this algorithm is tested on many multi-modal optimization problems. A new PSO variant (Mendes, Kennedy & Neves, 2004) is proposed by using the fully information around the particles to ameliorate the performance of PSO. The algorithm is simple and the solutions are better than some other PSO variants. An orthogonal learning PSO (Zhan, Zhang, Li & Shi, 2011) is developed by introducing into an orthogonal learning scheme into PSO to select the efficient exemplars. The more detail review of PSO can be found in reference (Jain, Nangia & Jain, 2018). To mimic the knowledge

* Corresponding author.
E-mail address: zfemail@163.com (F. Zou).

https://doi.org/10.1016/j.eswa.2022.117118
Received 15 June 2021; Received in revised form 21 January 2022; Accepted 28 March 2022
Available online 31 March 2022

(a) The ripe phase of seeds (b) The spread phase of seeds (c) The landing phase of seeds

Fig. 1. The different phases of seeds.

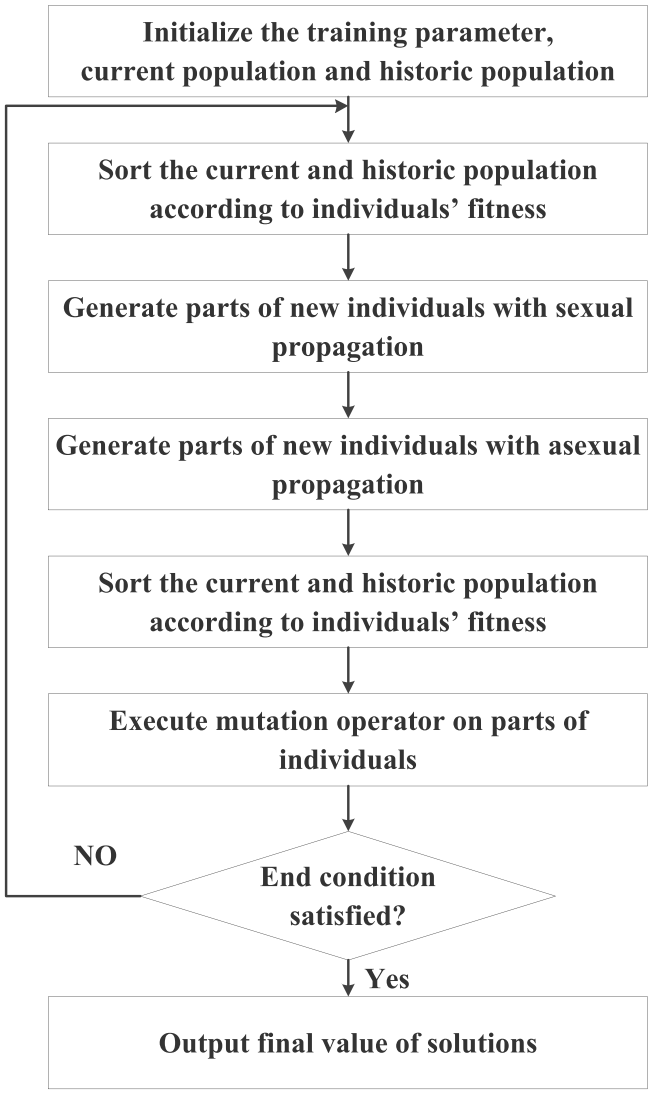

Fig. 2. The basic framework of POA.

sharing character between teacher and students in a class, teaching–learning-based optimization (TLBO) was developed in 2011 (Rao, Savsani & Vakharia, 2011). The algorithm was successfully used in many optimization fields for its simple structure and fewer parameters in updating equations. For the algorithm has zero bias composition, the performance of it is limited when the optimization solution is not zero. To ameliorate the performance of basic TLBO, some improved TLBO variants were developed. ETLBO (Rao & Patel, 2012) uses several teachers to guide optimization for learners, where the learners in the class can learn knowledge for different teacher so that the convergence speed and robustness of TLBO can be improved. DGSTLBO (Zou, Wang, Hei, Chen & Yang, 2014) was proposed by using grouping scheme, where each learner learns knowledge from the individuals of its neighborhood to improve the local convergence ability of the learner. Interesting readers can find the detail reviews of TLBO in reference (Zou, Chen & Xu, 2019). In addition, differential evolution (DE) (Storn & Price, 1997) and backtracking search optimization algorithm (BSA) (Civicioglu, 2013) are active in the optimization fields. Some variants are also developed to ameliorate the performance of these basic algorithms. jDE (Brest, Greiner, Boskovic, Mernik & Zumer, 2006) introduced a self-adapting parameter control method into DE, which is demonstrated to be significantly superior to DE. Self-adaptive DE (SaDE) was developed by introducing self-adaptive trial vector generation strategies and parameter control methods (Qin, Huang & Suganthan, 2009). The experimental results show that the convergence speed and reliability of solutions are greatly improved. Some issues and reviews of adaptive differential evolution were shown in reference (Das & Suganthan, 2011). BSA has some homologous characters of DE, but the history information is appropriately used in updating the positions of individuals. To ameliorate the convergence and reliability performance of BSA, two improved BSA variants called LBSA (Chen, Zou, Lu & Wang, 2017) and KLBSA (Chen, Zou, Lu & Li, 2019) are developed. In recent years, some new algorithms, such as fireworks algorithm (FWA)(Tan and Zhu, 2010) and sine–cosine optimization algorithm (SCA) (Mirjalili, 2016) are also developed in the optimization fields to mimic the explosion process of fireworks and use the sine and cosine function in renew process of individuals, respectively..

Though the above-mentioned algorithms play very important role in intelligence optimization domain, they still have some limitations. For example, the performance of some algorithms depends on the parameters. However, the exact parameters are very difficult to determine for some optimization problems. Some algorithm has zero bias trend, which means that the efficiency is high when the global optimal solution is zero. However, if the theoretical optimal solution is not zero, the efficiency of these algorithms are not ideal. As we know, the optima of complex optimization problems are usually not at zero in real problems. Designing efficient intelligence optimization algorithms with simple structure and easy operators for nonzero optimal solution problem is a hot research direction in intelligence optimization field.

Motivated by these considerations, in this paper, a novel poplar optimization algorithm (POA) is developed to deal with numerical optimization problems. The main contributions of this work are shown as following: (1) We firstly adopt a poplar optimization algorithm by mimicking the propagation mechanism of poplar. POA includes three process: the asexual propagation process for mimicking the plant cutting process of poplar with current and history information, the sexual propagation process for mimicking the seeds spreading process of poplar with the power of wind, and the simple mutation process for increasing the diversity of the population. (2) The performance of POA is tested on 55 functions from the CEC2005 test suite and the CEC2017 test suites with different features. The obtained results indicate that POA is competitive and even has its superiority in some cases.

The remaining parts of the paper are described as follows. In Section

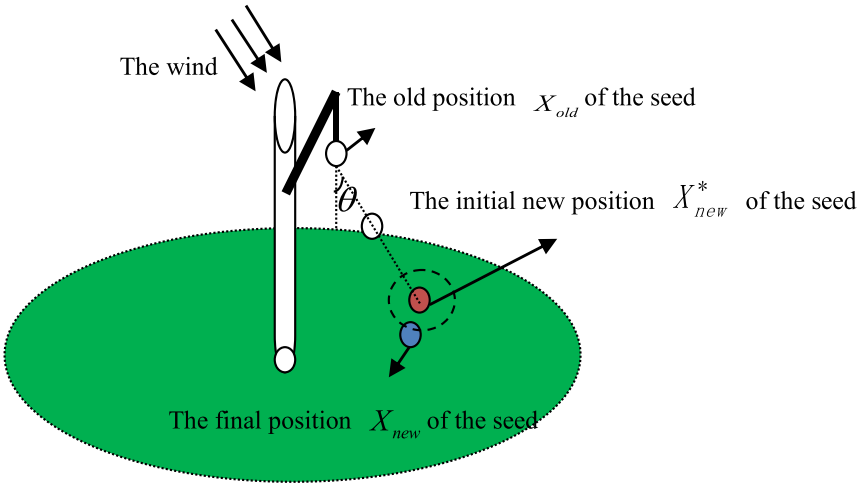

Fig. 3. The falling process of the seed.

2, we provide a brief introduction of the propagation mechanism of poplar. The designing procedure of POA is described in Section 3. Section 4 presents the experiments and testing results of different algorithms along with the statistical tests. Section 5 describes and evaluates the segmentation results on different benchmark images. Some conclusions and future works are given in Section 6.

2. The propagation mechanism of poplar

There are more than 100 species of poplar, and their propagation model can be roughly divided into two types: sexual propagation and asexual reproduction with plant cutting technology. In sexual propagation, adult poplars produce seeds, the ripe capsules cracked and the seeds drift to different places with poplar catkins under the power of wind. As shown in Fig. 1(a), the seeds of poplar are wrapped in the capsules with some poplar catkins around them, when the seeds are ripe, they will fall to the ground with external force, and the main force often comes from wind. The landing positions of the seeds determined by the direction and strength of the wind; it is also determined by the height of the seeds. When the seeds fall to the environment which suits for sprouting, the new seedling will be produced and then form a new poplar. Asexual propagation of poplar is usually done by plant cutting. The general operation of plant cutting is to cut a branch from the parent plant and insert it into the wet soil. The stronger branch of the parent poplar is often chosen as the cutting plant branch. The cutting branch can produce new roots, stems and or both by some asexual propagation technology, and thus is independent of the parent poplar and becomes a new poplar.

3. Poplar optimization algorithm

To clearly show the process of POA, Fig. 2 displays the framework. The main parts of POA are introduced in detail as follows.

3.1. Sexual propagation

The sexual propagation of poplar is that the new individual is propagated by the ripe seed. For poplar, the power of wind is usually the main external factor for determining the position of the seeds. The falling process of the seed with the power of wind is shown in Fig. 3.

In Fig. 3, the angle between the wind and the vertical direction isθ. To simplify the algorithm, assume each tree has only one ripe seed and the position of tree is the same as it of the seed. The old position of the tree or the seed is denoted byX_{old}. In real world, many undetermined factors, such as the gravity, the power of the wind and other external factors might affect the falling process of the seed. The falling trail of a seed is usually a complex curve. The precise position of the seed is difficult to be calculated. To simplify the determine process of the new position of the seed, the initial falling process of the seed is regarded as an oblique line. As shown in Fig. 3, the initial loading position X^*_{new} of the seed is expressed by the red circle. Considering the effects of other factors, the final position X_{new} of the seed is expressed by a blue circle. The position X^*_{new} of the seed is displayed as follow.

$$X^*_{new}(i,j) = X_{old}(i,j) + H(i)*tan(\theta) \tag{1}$$

where, $H(i)$ is height of i-th tree. For the seed is contained in the poplar catkin, the poplar catkin might change its position by the power of wind or other factors when it landing to the ground. This process is very complex. In POA, this process is simplified with a chaos factor. Eq.2 is the final new position of the i-th poplar.

$$X_{new}(i,j) = X_{old}(i,j) + r_1*H(i)*tan(\frac{\pi}{rr}*r_2)*(1.0 + F(i)*cos(r_3*\pi)) \tag{2}$$

In POA, sexual propagation is adopted by the former N*a individuals for the sexual propagation ability of the good poplar might strong. In Eq.2,$i = 1, 2, \cdots, a*N$, N is the scale of the population. r_1 and r_3 are the random numbers in the range [0, 1]. r_2 is a random number in the range [-1, 1], it is shown in Eq.3.

$$r_2 = 2.0*(rand(.) - 0.5) \tag{3}$$

$X_{new}(i,j)$ and $X_{old}(i,j)$ are the gens of current generation and previous generation.$F(i)$ is the chaos factor, a simple model with logistic mapping is chosen in Eq.4.

$$F^{k+1}(\mathrm{i}) = \mu*F^k(\mathrm{i})*(1.0 - F^k(\mathrm{i})) \tag{4}$$

$F^k(\mathrm{i})$ and $F^{k+1}(i)$ are the chaos factor of the i-th poplar in the k-th and the $(k+1)$-th iterations.$\mu = 4.0$. If X_{old} is worse thanX_{new}, then X_{old} is replaced byX_{new}.

To make the updating amplitude relatively large in the initial evolution phase and small in the later evolution period, a variation factor rr is adopt to modify the angle θ with linear descend model, the formulation of rr is shown in Eq.5.

$$rr = 10.0 - 8.0*(maxgen - gen)/maxgen \tag{5}$$

The next task is to determine the height of i-th poplar. In our work, the height of a tree is determined by the fitness of all trees. Assume the boundary value of variables is denoted byH_{max}. The height of i-th tree is designed in Eq.6.

$$H(i) = \alpha^* H_{max}{}^*(\frac{1.0/fit(i)}{\sum_{i=1}^{N} 1.0/fit(i) + \varepsilon}) \tag{6}$$

where, α is a constant in the range [0,1]. $fit(i)$ is the fitness of i-th individual, if$\sum_{i=1}^{N} 1.0/fit(i) = 0$, then ε is a small constant which makes the algorithm not overflow, otherwise ε is zero. Eq.6 indicates that the height of the individual has adaptive character. For minimal optimization problem, the height of the individual with small fitness value is relatively large.

To clearly show the sexual propagation process of POA for solving minimal optimization problem, the steps of it are displayed in algorithm 1.

Algorithm 1:

```
According to their performance, all individuals are sorted in ascending order;
  For i = 1:α*N
    Calculate the chaos factor of each individual according to equation (4);
    Calculate the height of each individual according to equation (6);
    Generate random number r1 and r3;
    For j = 1:D
      Determine rr according to equation (5);
      Calculate r2 according to equation (3);
      Renew the position the i-th poplar according to equation (2);
      Execute boundary conditions of all variables;
    Endfor
    EvaluateX_new ;
    If X_old is worse thanX_new, then X_old is replaced byX_new;
  Endfor
```

3.2. Asexual reproduction

For the remainder individuals, the new positions of them are generated by asexual reproduction. The branches which are used for plant cutting come from the randomly selected individuals of the former part. For the gens of the plant cutting branch is the same as its father, if the gens of it is not change, the new position of it is almost the same as its father. This reproduction method is not benefit for generating new individuals. The diversity of the population might lose quickly. To balance the convergence and diversity character, the best individual of current population and the history information of the selected good individual are used in asexual reproduction process. The best individual plays a guiding role, this operator benefits for convergence. The history information benefits for the diversity. To balance explore and exploit abilities of POA, an adaptive method with considering the fitness of different individuals are designed in the algorithm. The detail process of asexual reproduction is shown as follow.

First of all, a poplar kk in former part of the population is randomly chosen for updating k-th poplar. For individuals with different characteristics have different effects on the updated individuals, an adaptive factor C according to the fitness of the best individuals in current iteration and its historical information is designed to adjust the effect proportion of them. The coefficient c is shown in Eq.7.

$$C = f_{best}/(f_{best} + f_{oldp,kk}) \tag{7}$$

where, f_{best} is the best fitness of current popular, $f_{oldp,kk}$ is the history fitness of the kk-th individual in historic population.

The new position of the k-th poplar is updated by Eq.8.

$$X_{new}(k,j) = X_{old}(kk,j) + (1.0 - C)^*(X_{best}(1,j) - X_{old}(k,j)) + C^*(X_{oldp}(kk,j) - X_{oldp}(k,j)) \tag{8}$$

In Eq.8,$k = a\%*N + 1, \cdots, N$, kk is a random number in the range [1, $a\%*N$]. $X_{new}(k,j)$ is the position of the k-th poplar in the j-th dimension after updating procedure. $X_{old}(kk,j)$ and $X_{old}(k,j)$ are the position of the kk-th and k-th poplar in the j-th dimension of previous generation. $X_{best}(1,j)$ is the position of the best poplar in the j-th dimension of current generation. $X_{oldp}(kk,j)$ and $X_{oldp}(k,j)$ are the history positions of the kk-th and the k-th poplars at the j-th dimension in historic population.

As it used in BSA, the historic population is formed as follows. The initial history population is generated randomly. It is renewed with Eq. (9) and Eq.10.

$$oldP = \begin{cases} P & if\ r_4 < r_5 \\ oldp & otherwise \end{cases} \tag{9}$$

$$oldP := permuting(oldP) \tag{10}$$

In the equations, P is the evolution population, and $oldP$ is the history population. r_4 and r_5 are the random numbers in the range [0,1]. Eq.10 shows that the individuals in old population will be permuted after it is generated.

The steps of asexual reproduction are displayed in Algorithm 2.

Algorithm 2:

```
Update historic population according to equations (9) and (10);
  For k = a%*N + 1:N
    kk = ceil(rand*a%*N);
    Calculate the value of C according to equation (7);
    For j = 1:D
      Renew the position k-th poplar according to equation (8);
      Execute boundary conditions of all variables;
    Endfor
    EvaluateX_new;
    if X_old is worse thanX_new, acceptX_new;
  Endfor
```

3.3. Mutation

There are many mutation strategies in evolutionary computation. A classical mutation method in BSA is adopted. For minimal value problem, the individuals are sorted in ascending order according their fitness. For k-th individual in the latter part, the historic information of it is used in mutation. This method mimics, the mutation operator is shown in Eq. (11).

$$X^*(k,d) = X(k,d) + \beta^* rand(.)^*(X_{oldp}(k,d) - X(k,d)) \tag{11}$$

where, $X^*(k,d)$ and $X(k,d)$ are the gens of the new and old positions of k-th poplar at the d-th dimension, β is an updating coefficient, it determines the updating step size to the individual. $X_{oldp}(k,d)$ is the gens of the k-th individual at the d-th dimension in the historic population. In the algorithm,$k = b^*N + 1, \cdots, N$, b is a constant which is used to determine the number of mutation individuals. In POA the individuals at the bottom are mutated.

Table 1
Comparison of POA with different parameters.

Parameter	$a = 0.2$		$a = 0.4$		$a = 0.6$		$a = 0.8$		$a = 1.0$	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
F2	2.9000E + 01	3.9598E + 01	**0.0000E + 00**	0.0000E + 00	**0.0000E + 00**	0.0000E + 00	**0.0000E + 00**	0.0000E + 00	2.8500E + 01	4.0305E + 01
F4	**1.3946E + 01**	1.4072E + 01	4.7589E + 01	3.4466E + 01	2.1242E + 01	2.7881E + 00	3.5504E + 01	4.4447E + 01	4.3750E + 01	3.4747E + 01
F8	3.5356E + 01	6.9350E-01	**2.9434E + 01**	5.8689E-01	3.9375E + 01	8.0910E-01	4.4276E + 01	6.3318E + 00	4.3506E + 01	1.4455E + 01
F12	4.0256E + 04	4.3879E + 04	9.2424E + 03	2.0626E + 03	1.7334E + 04	1.4719E + 04	**6.6729E + 03**	4.1779E + 03	7.0774E + 03	1.8871E + 03
F16	6.2118E + 02	4.1794E + 01	5.9905E + 02	2.8056E + 02	**4.2033E + 02**	2.3865E + 02	7.4572E + 02	6.5321E + 01	5.5066E + 02	3.7311E + 02
F21	2.6373E + 02	1.6543E + 01	2.2932E + 02	1.3577E + 00	2.3485E + 02	2.9057E + 00	**2.2787E + 02**	6.7396E + 00	2.3512E + 02	7.6417E-01
F24	4.9273E + 02	2.8574E + 01	4.8099E + 02	1.3648E + 01	**4.6478E + 02**	2.7965E + 00	4.8985E + 02	2.3763E + 01	4.7290E + 02	1.4534E + 01
Parameter	$b = 0.2$		$b = 0.4$		$b = 0.6$		$b = 0.8$		$b = 1.0$	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
F2	3.3500E + 01	4.7376E + 01	2.3500E + 01	6.3640E + 00	**0.0000E + 00**	0.0000E + 00	**0.0000E + 00**	0.0000E + 00	1.0208E + 06	5.7076E + 05
F4	2.9281E + 01	4.1409E + 01	**1.9941E + 00**	2.8182E + 00	2.2937E + 01	3.1441E + 01	4.9459E + 00	9.3045E-01	9.9205E + 01	1.9452E + 01
F8	**3.5321E + 01**	4.9248E + 00	3.9301E + 01	1.6181E + 01	3.9301E + 01	1.3367E + 01	3.6287E + 01	2.0979E-01	7.2322E + 01	1.5388E + 01
F12	7.1573E + 03	4.3230E + 03	5.3177E + 03	1.1164E + 03	**4.2853E + 03**	3.9583E + 03	1.0848E + 04	2.0931E + 03	5.5856E + 06	1.7367E + 06
F16	**2.6801E + 02**	3.1479E + 00	6.0661E + 02	1.1285E + 02	5.5077E + 02	1.7801E + 02	6.8419E + 02	2.2162E + 02	9.1095E + 02	1.2030E + 02
F21	2.4002E + 02	1.7124E + 01	**2.2937E + 02**	1.3914E + 01	2.4118E + 02	6.7476E + 00	2.3774E + 02	4.5943E + 00	2.6870E + 02	1.4180E + 01
F24	**4.5047E + 02**	6.0380E + 00	4.7210E + 02	2.7589E + 01	4.9942E + 02	1.5120E + 01	4.7539E + 02	1.4259E + 00	5.3514E + 02	2.3307E + 01
Parameter	$\alpha = 0.2$		$\alpha = 0.4$		$\alpha = 0.6$		$\alpha = 0.8$		$\alpha = 1.0$	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
F2	4.0000E + 00	5.6569E + 00	**0.0000E + 00**	0.0000E + 00	9.0000E + 00	1.2728E + 01	1.1500E + 01	1.6263E + 01	5.1367E + 16	1.0677E + 16
F4	**2.1316E + 00**	2.6274E + 00	2.2952E + 00	2.6660E + 00	2.3157E + 01	2.2170E + 00	4.4312E + 01	3.3355E + 01	1.1662E + 01	1.6308E + 01
F8	3.6523E + 01	6.6250E + 00	3.4022E + 01	8.7165E + 00	**2.6202E + 01**	6.0993E + 00	5.0572E + 01	8.9654E + 00	1.1533E + 02	6.5218E + 01
F12	9.2989E + 03	8.6852E + 02	1.7985E + 04	1.6479E + 04	**7.0828E + 03**	2.4698E + 03	1.7630E + 04	1.9531E + 03	3.0944E + 04	2.0729E + 04
F16	6.2109E + 02	6.9505E + 01	**4.2242E + 02**	8.1423E + 01	5.5896E + 02	1.7573E + 02	7.7618E + 02	6.2352E + 02	1.0446E + 03	8.4176E + 01
F21	2.4270E + 02	1.6363E + 00	2.4409E + 02	1.3793E + 01	2.3572E + 02	9.8199E + 00	**2.3002E + 02**	2.4171E + 00	3.2664E + 02	1.2007E + 01
F24	**4.6286E + 02**	3.6091E + 00	4.7032E + 02	2.9070E + 00	4.7226E + 02	1.5341E + 01	5.0837E + 02	4.3402E + 01	5.6889E + 02	3.0991E + 01
Parameter	$\beta = 0.8$		$\beta = 1.0$		$\beta = 1.2$		$\beta = 1.4$		$\beta = 1.6$	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
F2	**0.0000E + 00**	0.0000E + 00	2.0000E + 01	2.8284E + 01	3.0500E + 01	4.3134E + 01	**0.0000E + 00**	0.0000E + 00	**0.0000E + 00**	0.0000E + 00
F4	**4.1607E + 00**	9.7221E-02	3.2075E + 01	4.5315E + 01	6.1339E + 01	3.9286E + 00	3.4095E + 01	4.2458E + 01	2.9338E + 01	4.1490E + 01
F8	5.9684E + 01	1.2294E + 01	4.1021E + 01	1.0851E + 00	3.9301E + 01	7.7388E + 00	**2.6795E + 01**	6.3659E + 00	3.3712E + 01	6.8710E + 00
F12	**9.7002E + 03**	1.6233E + 03	1.2595E + 04	7.3887E + 03	1.1962E + 04	1.4245E + 03	2.5350E + 04	2.3488E + 04	1.4679E + 04	4.0035E + 03
F16	7.5620E + 02	8.2397E + 01	**3.8020E + 02**	1.8171E + 02	5.4623E + 02	3.3798E + 02	5.7963E + 02	4.6804E + 01	6.0180E + 02	2.8552E + 02
F21	2.3832E + 02	6.7970E + 00	2.3663E + 02	6.6520E + 00	2.4419E + 02	1.5051E + 01	**2.3289E + 02**	1.2990E + 01	2.3780E + 02	4.5024E + 00
F24	4.8950E + 02	8.2484E + 00	4.8719E + 02	1.7244E + 00	4.7556E + 02	1.0148E + 01	**4.5396E + 02**	8.2798E-01	4.7479E + 02	3.4517E + 00

Table 2
Comparison of POA with six basic algorithms for CeC2005 test suite.

Algorithm	Indexes	F1	F2	F3	F4	F5
ABC	Mean	4.9616E-16‡	1.2813E + 04‡	1.2364E + 07‡	3.3454E + 04‡	1.0442E + 04‡
	Std.	5.5097E-17	2.7913E + 03	3.3654E + 06	3.1786E + 02	1.9944E + 03
CBO	Mean	7.2996E-27‡	1.3219E + 00‡	8.1667E + 06‡	2.8672E + 03‡	6.0259E + 03‡
	Std.	3.0609E-27	1.0595E + 00	5.8841E + 05	9.8532E + 02	1.7678E + 03
FWA	Mean	2.6979E + 04‡	3.3596E + 04‡	1.5285E + 08‡	2.5582E + 04‡	2.0126E + 04‡
	Std.	2.9697E + 03	3.6766E + 03	1.5025E + 07	2.3393E + 03	7.8801E + 02
BSA	Mean	8.0589E-17‡	1.7674E + 03‡	4.3172E + 06‡	1.1938E + 04‡	3.1841E + 03‡
	Std.	6.0862E-18	6.5685E + 02	1.6469E + 06	3.9716E + 03	2.6299E + 02
SCA	Mean	3.6033E + 04‡	4.8712E + 04‡	8.0835E + 08‡	1.2605E + 05‡	2.9970E + 04‡
	Std.	3.4498E + 03	8.6895E + 03	4.3595E + 08	5.5273E + 04	5.2017E + 03
TLBO	Mean	6.9877E-25‡	2.5706E-03‡	1.1191E + 06‡	1.0366E + 03‡	3.3060E + 03‡
	Std.	9.5084E-25	4.4093E-03	6.5929E + 05	9.6563E + 02	8.8286E + 02
POA	Mean	**0.0000E + 00**	**3.6014E-14**	**1.4729E + 05**	**6.3186E-02**	**7.2858E + 02**
	Std.	0.0000E + 00	1.8977E-14	7.8765E + 04	8.4178E-02	3.4038E + 02
Rank		1	1	1	1	1
Algorithm	Indexes	F6	F7	F8	F9	F10
ABC	Mean	1.7986E + 00‡	5.0114E-01	2.0902E + 01	**9.3732E-13**	3.1383E + 02
	Std.	1.4507E + 00	9.4387E-02	1.2002E-01	1.3905E-12	6.0538E + 01
CBO	Mean	6.7410E + 02‡	1.4784E-02	2.0705E + 01	9.7506E + 01‡	7.0310E + 01
	Std.	4.0292E + 02	8.4812E-03	4.4801E-01	9.4913E + 00	6.3966E + 00
FWA	Mean	6.4987E + 09‡	3.5394E + 03	2.0672E + 01	1.1999E + 02‡	1.8636E + 02
	Std.	1.2614E + 09	1.3097E + 02	1.3897E-01	1.0491E + 01	4.7728E + 01
BSA	Mean	1.2784E + 02‡	4.7356E-01	2.0984E + 01	2.6320E + 00	7.5926E + 01
	Std.	7.5179E + 01	1.6105E-01	5.2083E-02	7.2142E-01	5.9306E + 00
SCA	Mean	1.0408E + 10‡	1.6309E + 03	2.0928E + 01	3.5415E + 02‡	5.2417E + 02
	Std.	3.9795E + 09	3.2501E + 02	1.0760E-02	6.1148E + 01	3.4528E + 01
TLBO	Mean	5.3365E + 01‡	3.1716E-02	2.0948E + 01	8.9878E + 01‡	9.0417E + 01
	Std.	3.3443E + 01	3.3959E-02	2.2658E-02	1.5231E + 01	3.7448E + 01
POA	Mean	**7.1142E-02**	**1.2301E-02**	**2.0681E + 01**	3.2834E + 01	**4.2453E + 01**
	Std.	1.1535E-01	1.0724E-02	5.2665E-02	6.8933E + 00	3.1960E + 00
Rank		1	1	1	3	1
Algorithm	Indexes	F11	F12	F13	F14	F15
ABC	Mean	2.8738E + 01‡	1.5646E + 04‡	**1.4636E + 00**†	1.2822E + 01‡	**5.0976E + 00**†
	Std.	3.2865E + 00	5.5249E + 02	1.1100E-01	1.1886E-01	2.3619E + 00
CBO	Mean	4.0091E + 01‡	5.8995E + 04‡	5.6501E + 00‡	1.4253E + 01‡	8.7993E + 02‡
	Std.	1.0869E + 00	6.4533E + 04	1.6853E + 00	2.4184E-01	7.4320E + 01
FWA	Mean	3.2515E + 01‡	1.1711E + 05‡	1.1036E + 01‡	1.3112E + 01‡	5.0463E + 02‡
	Std.	6.4245E + 00	5.0563E + 04	1.8086E-01	3.1553E-01	1.0273E + 01
BSA	Mean	2.8339E + 01‡	1.9500E + 04‡	2.6380E + 00†	1.3093E + 01‡	8.5854E + 01†
	Std.	4.5421E-01	1.1373E + 04	3.6658E-01	7.3462E-02	2.0641E + 01
SCA	Mean	4.0512E + 01‡	1.0835E + 06‡	3.1036E + 02‡	1.4269E + 01‡	9.0212E + 02‡
	Std.	2.8528E-01	1.3838E + 05	4.3309E + 02	2.1677E-01	2.4313E + 01
TLBO	Mean	3.6615E + 01‡	**8.1433E + 03**†	4.1598E + 00‡	1.2827E + 01‡	5.0583E + 02‡
	Std.	1.0298E + 00	2.5726E + 03	8.1942E-01	9.7795E-02	1.0191E + 02
POA	Mean	**1.5341E + 01**	9.9315E + 03	2.6760E + 00	**1.2686E + 01**	3.2129E + 02
	Std.	4.6955E + 00	6.9594E + 03	1.0538E-01	2.8223E-01	6.8518E + 01
Rank		1	2	3	1	3
Algorithm	Indexes	F16	F17	F18	F19	F20
ABC	Mean	3.3601E + 02‡	4.8978E + 02‡	9.2177E + 02†	9.1587E + 02‡	9.2004E + 02‡
	Std.	4.1627E + 01	2.9409E + 01	7.3516E + 00	2.6870E + 00	3.2532E + 00
CBO	Mean	2.4512E + 02‡	3.8248E + 02‡	**8.3531E + 02**†	**8.3178E + 02**†	**8.3171E + 02**†
	Std.	1.4593E + 02	8.9870E + 01	3.5166E-01	2.2764E + 00	2.7890E + 00
FWA	Mean	2.9808E + 02‡	1.7354E + 02‡	9.7226E + 02‡	9.2680E + 02‡	9.2840E + 02‡
	Std.	1.3785E + 02	4.9092E + 01	1.3830E + 01	2.0039E + 01	2.5911E + 01
BSA	Mean	**1.0282E + 02**†	1.9876E + 02‡	9.1568E + 02†	8.7407E + 02†	8.7544E + 02†
	Std.	1.1314E + 01	2.0300E + 01	7.3423E-01	6.4148E + 01	6.5435E + 01
SCA	Mean	6.7749E + 02‡	6.7629E + 02‡	1.1890E + 03‡	1.1658E + 03‡	1.1554E + 03‡
	Std.	1.2497E + 02	1.0171E + 02	9.7350E + 01	7.4636E + 01	6.6300E + 01
TLBO	Mean	3.1166E + 02‡	1.4190E + 02‡	9.3935E + 02‡	9.2682E + 02‡	9.2831E + 02‡
	Std.	1.7043E + 02	3.8801E + 01	3.4111E + 01	2.6636E + 00	1.6218E + 01
POA	Mean	2.3617E + 02	**1.1262E + 02**	9.2706E + 02	9.0701E + 02	9.0678E + 02
	Std.	2.3452E + 02	5.2341E + 01	3.3943E + 01	1.6880E + 00	2.0324E + 00
Rank		2	1	4	3	3
Algorithm	Indexes	F21	F22	F23	F24	F25
ABC	Mean	**4.4699E + 02**†	1.0851E + 03‡	**5.2878E + 02**†	1.1513E + 03‡	1.3452E + 03‡
	Std.	4.5914E + 01	2.7538E + 01	9.3264E + 00	9.0507E + 01	1.2111E + 01
CBO	Mean	8.7159E + 02‡	**6.3350E + 02**†	8.7802E + 02‡	2.1361E + 02†	**2.1422E + 02**†
	Std.	6.2449E + 00	1.9398E + 02	6.0435E + 00	6.7034E-01	2.4399E + 00
FWA	Mean	1.0693E + 03‡	8.5456E + 02†	9.9932E + 02‡	3.9999E + 02†	3.1644E + 02†
	Std.	1.4532E + 02	4.0859E + 01	1.9308E + 01	1.7346E + 02	2.0052E + 02
BSA	Mean	5.0000E + 02	9.4683E + 02‡	5.3416E + 02†	**2.0000E + 02**†	1.0271E + 03‡
	Std.	4.2727E-13	1.5413E + 01	1.1814E-04	1.8631E-12	5.8880E + 00
SCA	Mean	1.3237E + 03‡	1.3565E + 03‡	1.3399E + 03‡	1.3636E + 03‡	1.2028E + 03‡
	Std.	2.0736E + 01	6.4808E + 02	6.6548E + 01	2.6840E + 01	2.2000E + 02

(*continued on next page*)

Table 2 (*continued*)

Algorithm	Indexes	F1	F2	F3	F4	F5
TLBO	Mean	7.4315E + 02‡	9.3769E + 02‡	9.3732E + 02‡	2.0000E + 02†	1.0494E + 03‡
	Std.	4.2115E + 02	9.3322E + 00	2.6557E + 02	1.4246E-10	5.8099E + 01
POA	Mean	5.0000E + 02	9.0672E + 02	7.2175E + 02	9.6138E + 02	9.7326E + 02
	Std.	1.6573E-13	4.4219E + 01	3.2490E + 02	5.8040E + 00	5.8604E + 00
Rank		2	3	3	5	3

Table 3
Comparison of POA with eight variants of basic algorithms for CEC2005 test suite.

Algorithm	Indexes	F1	F2	F3	F4	F5
jDE	Mean	**0.0000E + 00**~	1.0612E-03‡	5.3699E + 05‡	4.9414E + 01‡	2.1224E + 03‡
	Std.	0.0000E + 00	6.9596E-04	2.8984E + 05	6.2228E + 01	9.5512E + 02
SaDE	Mean	**0.0000E + 00**~	5.4114E-03‡	9.6254E + 05‡	3.5678E + 02‡	2.5121E + 03‡
	Std.	0.0000E + 00	4.2511E-03	5.6541E + 05	4.3512E + 02	3.0884E + 02
PSOwFIPS	Mean	1.5378E-06‡	4.8620E + 02‡	1.2624E + 07‡	2.0944E + 03‡	2.6224E + 03‡
	Std.	1.2958E-06	5.6955E + 01	3.3920E + 06	1.0164E + 03	8.6727E + 01
FDRPSO	Mean	1.6060E + 02‡	1.1046E + 02‡	2.9873E + 06‡	1.6714E + 03‡	6.2083E + 03‡
	Std.	1.9567E + 02	9.5767E + 01	1.9126E + 06	2.8339E + 02	5.9307E + 02
LBSA	Mean	**0.0000E + 00**~	5.7683E-02‡	1.0121E + 06‡	3.6275E + 02‡	3.6222E + 03‡
	Std.	0.0000E + 00	3.4178E-02	1.5939E + 05	3.0239E + 02	2.3613E + 02
MBSA	Mean	**0.0000E + 00**~	3.5566E-05‡	2.9823E + 05‡	1.0128E + 02‡	2.8037E + 03‡
	Std.	0.0000E + 00	3.4710E-05	8.0607E + 04	1.2327E + 02	2.8923E + 02
ETLBO	Mean	3.9816E-27‡	7.1416E-05‡	1.2755E + 06‡	9.7688E + 02‡	2.8158E + 03‡
	Std.	4.3164E-27	7.1797E-05	3.8109E + 05	1.3750E + 03	4.1399E + 02
DGSTLBO	Mean	4.7098E-10‡	3.5583E + 02‡	7.2191E + 06‡	3.2777E + 02‡	9.4389E + 03‡
	Std.	4.6474E-10	2.5326E + 02	2.0677E + 06	3.2811E + 02	2.6640E + 03
POA	Mean	**0.0000E + 00**	**3.6014E-14**	**1.4729E + 05**	**6.3186E-02**	**7.2858E + 02**
	Std.	0.0000E + 00	1.8977E-14	7.8765E + 04	8.4178E-02	3.4038E + 02
Rank		1	1	1	1	1
Algorithm	Indexes	F6	F7	F8	F9	F10
jDE	Mean	5.9252E + 00‡	1.3942E-02‡	2.0955E + 01‡	**0.0000E + 00**†	6.5343E + 01‡
	Std.	5.1499E + 00	9.2887E-03	4.8902E-02	0.0000E + 00	1.3400E + 01
SaDE	Mean	6.0196E + 01‡	**9.8717E-03**†	2.0948E + 01‡	**0.0000E + 00**†	9.3014E + 01‡
	Std.	3.0614E + 01	1.1210E-05	5.1602E-02	0.0000E + 00	4.6223E + 00
PSOwFIPS	Mean	4.9096E + 01‡	3.9032E-01‡	2.0988E + 01‡	6.1567E + 01‡	1.8575E + 02‡
	Std.	3.9247E + 01	1.3755E-01	1.8286E-02	4.1517E + 00	4.8495E + 00
FDRPSO	Mean	1.6309E + 08‡	1.3495E + 01‡	2.0885E + 01‡	4.1361E + 01‡	1.0620E + 02‡
	Std.	2.2919E + 08	3.7714E + 00	8.4370E-02	1.7121E + 01	2.5574E + 01
LBSA	Mean	6.7429E + 01‡	4.7516E-02‡	2.0969E + 01‡	**0.0000E + 00**†	7.4661E + 01‡
	Std.	4.1592E + 01	1.7722E-02	5.6307E-02	0.0000E + 00	8.7580E + 00
MBSA	Mean	4.5518E + 00‡	2.2876E-02‡	2.0956E + 01‡	**0.0000E + 00**†	6.7241E + 01‡
	Std.	2.7152E + 00	2.2542E-02	2.4440E-02	0.0000E + 00	1.1846E + 01
ETLBO	Mean	4.1178E + 01‡	2.1777E-02‡	2.0995E + 01‡	7.9597E + 01‡	1.1568E + 02‡
	Std.	2.7629E + 01	2.8364E-02	3.0785E-02	3.9686E + 01	2.2411E + 01
DGSTLBO	Mean	2.9194E + 04‡	1.5052E + 01‡	2.1023E + 01‡	1.0613E + 02‡	1.3880E + 02‡
	Std.	2.6454E + 04	1.4037E + 01	3.4438E-02	2.6962E + 01	4.0733E + 01
POA	Mean	**7.1142E-02**	1.2301E-02	**2.0681E + 01**	3.2834E + 01	**4.2453E + 01**
	Std.	1.1535E-01	1.0724E-02	5.2665E-02	6.8933E + 00	3.1960E + 00
Rank		1	2	1	5	1
Algorithm	Indexes	F11	F12	F13	F14	F15
jDE	Mean	3.3446E + 01‡	3.2686E + 04‡	**1.7464E + 00**†	1.3339E + 01‡	**1.7863E + 02**†
	Std.	3.7077E + 00	6.4090E + 03	2.8753E-02	1.5666E-01	4.5972E + 01
SaDE	Mean	3.1335E + 01‡	4.4478E + 03†	4.5820E + 00‡	1.3097E + 01‡	3.3333E + 02‡
	Std.	2.3734E + 00	3.3397E + 03	2.2080E-01	7.0693E-02	5.7735E + 01
PSOwFIPS	Mean	2.7994E + 01‡	2.1881E + 04‡	1.3213E + 01‡	1.2831E + 01‡	3.1431E + 02†
	Std.	1.7627E + 00	7.4285E + 02	6.1760E-01	9.8382E-02	1.9612E + 01
FDRPSO	Mean	1.7856E + 01‡	4.0390E + 03†	3.8500E + 00‡	**1.1788E + 01**†	3.6908E + 02‡
	Std.	3.7507E + 00	5.9346E + 03	1.4510E + 00	7.4293E-01	2.0581E + 02
LBSA	Mean	2.5481E + 01‡	4.2677E + 03†	1.8098E + 00†	1.2582E + 01†	2.3349E + 02†
	Std.	4.0150E + 00	2.4394E + 03	2.1021E-01	2.8589E-01	2.0824E + 02
MBSA	Mean	2.9763E + 01‡	**1.3297E + 03**†	2.6413E + 00†	1.2870E + 01‡	4.0053E + 02‡
	Std.	6.9978E-01	1.2646E + 03	3.5689E-01	2.6037E-01	9.1319E-01
ETLBO	Mean	3.6572E + 01‡	1.5944E + 04‡	3.6648E + 00‡	1.3176E + 01‡	4.9168E + 02‡
	Std.	4.0508E + 00	1.0082E + 04	3.7358E-01	2.2576E-01	2.7778E + 01
DGSTLBO	Mean	2.4652E + 01‡	3.9817E + 04‡	7.9133E + 00‡	1.2389E + 01†	4.7666E + 02‡
	Std.	1.5687E + 00	1.2921E + 04	2.3762E + 00	2.8822E-02	6.2673E + 01
POA	Mean	**1.5341E + 01**	9.9315E + 03	2.6760E + 00	1.2686E + 01	3.2129E + 02
	Std.	4.6955E + 00	6.9594E + 03	1.0538E-01	2.8223E-01	6.8518E + 01
Rank		1	5	4	4	4
Algorithm	Indexes	F16	F17	F18	F19	F20
jDE	Mean	1.2554E + 02†	1.4006E + 02‡	**8.1679E + 02**†	**8.1884E + 02**†	**8.1846E + 02**†
	Std.	6.4625E + 01	2.8982E + 01	5.1668E-01	1.2643E + 00	1.7075E + 00
SaDE	Mean	1.3954E + 02†	1.6116E + 02‡	9.0923E + 02†	9.1239E + 02‡	9.1310E + 02‡

(*continued on next page*)

Table 3 (*continued*)

Algorithm	Indexes	F1	F2	F3	F4	F5
	Std.	7.4925E + 01	2.1711E + 01	2.8615E + 00	1.3527E + 00	2.9475E + 00
PSOwFIPS	Mean	2.6826E + 02‡	2.9490E + 02‡	8.3292E + 02†	8.3173E + 02†	8.3257E + 02†
	Std.	4.5587E + 01	6.6517E + 01	1.4795E + 00	1.4374E + 00	3.6872E-01
FDRPSO	Mean	3.6030E + 02‡	4.3593E + 02‡	9.4043E + 02‡	9.4541E + 02‡	9.1374E + 02‡
	Std.	2.0308E + 02	2.7731E + 02	2.5440E + 01	2.3888E + 01	1.4673E + 00
LBSA	Mean	1.3017E + 02†	2.1511E + 02‡	8.7835E + 02†	9.1643E + 02‡	9.1503E + 02‡
	Std.	4.4214E + 01	1.3409E + 02	6.8244E + 01	6.4884E + 00	1.7661E + 00
MBSA	Mean	**6.4692E + 01**†	2.1514E + 02‡	8.7768E + 02†	9.2250E + 02‡	9.2775E + 02‡
	Std.	2.0476E + 01	1.7941E + 02	6.7288E + 01	1.7380E + 00	1.0354E + 01
ETLBO	Mean	2.6054E + 02‡	3.2917E + 02‡	9.1868E + 02†	9.2328E + 02‡	9.4497E + 02‡
	Std.	2.1170E + 02	2.3238E + 02	7.6316E + 00	7.3784E + 00	1.1525E + 01
DGSTLBO	Mean	2.5267E + 02‡	2.5649E + 02‡	9.7221E + 02‡	9.8705E + 02‡	9.6625E + 02‡
	Std.	2.2231E + 02	6.0453E + 01	2.1564E + 01	2.6036E + 01	1.8488E + 01
POA	Mean	2.3617E + 02	**1.1262E + 02**	9.2706E + 02	9.0701E + 02	9.0678E + 02
	Std.	2.3452E + 02	5.2341E + 01	3.3943E + 01	1.6880E + 00	2.0324E + 00
Rank		5	1	7	3	3
Algorithm	Indexes	F21	F22	F23	F24	F25
jDE	Mean	8.6283E + 02‡	**5.0290E + 02**†	8.7258E + 02‡	2.1138E + 02†	2.1226E + 02†
	Std.	2.5662E + 00	4.2643E + 00	7.5887E + 00	8.7478E-01	9.9219E-01
SaDE	Mean	**5.0000E + 02**~	9.2835E + 02‡	**5.3416E + 02**†	2.0000E + 02†	9.9334E + 02‡
	Std.	0.0000E + 00	1.4143E + 01	2.6776E-04	3.4809E-14	5.9602E + 00
PSOwFIPS	Mean	5.0000E + 02~	5.3132E + 02†	5.3416E + 02†	2.1692E + 02†	**2.0722E + 02**†
	Std.	1.2753E-04	2.3778E + 00	2.8171E-04	1.5020E-01	1.2497E + 01
FDRPSO	Mean	1.0137E + 03‡	8.8343E + 02†	1.0369E + 03‡	9.3651E + 02†	1.1040E + 03‡
	Std.	1.0373E + 02	1.7274E + 01	2.0283E + 02	8.3628E + 00	1.4902E + 01
LBSA	Mean	5.0000E + 02~	9.3624E + 02‡	5.3558E + 02‡	2.0000E + 02†	9.9547E + 02‡
	Std.	2.1645E-13	2.5173E + 01	2.4545E + 00	1.7886E-12	1.2073E + 01
MBSA	Mean	5.0000E + 02~	9.3622E + 02‡	7.5095E + 02‡	**2.0000E + 02**†	9.8621E + 02‡
	Std.	2.0886E-13	5.4452E + 00	3.6453E + 02	0.0000E + 00	3.0932E + 00
ETLBO	Mean	9.6403E + 02‡	9.1000E + 02‡	1.0475E + 03‡	4.5905E + 02†	1.0076E + 03‡
	Std.	4.0223E + 02	4.0046E + 01	2.3229E + 02	4.4869E + 02	1.3977E + 01
DGSTLBO	Mean	7.9915E + 02‡	9.7433E + 02‡	1.1300E + 03‡	6.5310E + 02†	1.2503E + 03‡
	Std.	3.8359E + 02	2.9272E + 01	1.5247E + 02	4.8245E + 02	1.3996E + 01
POA	Mean	5.0000E + 02	9.0672E + 02	7.2175E + 02	9.6138E + 02	9.7326E + 02
	Std.	1.6573E-13	4.4219E + 01	3.2490E + 02	5.8040E + 00	5.8604E + 00
Rank		2	4	3	9	3

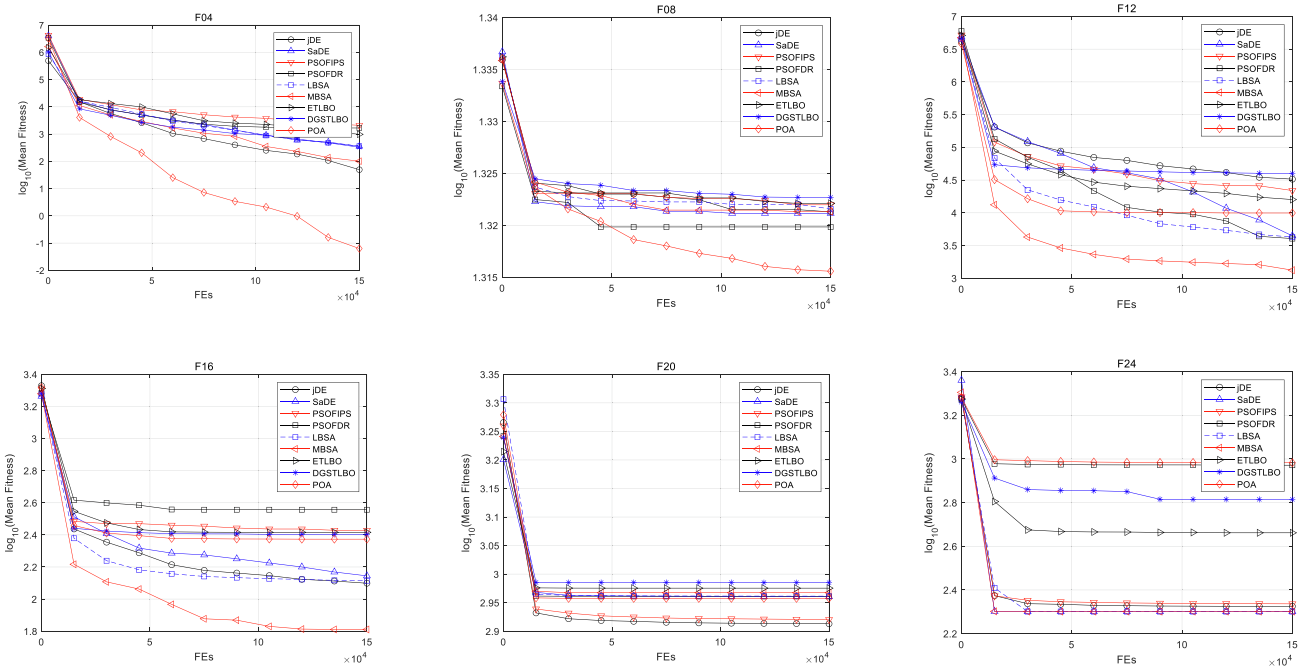

Fig. 4. Comparison of the performance curves using different algorithms.

Table 4
Comparison of POA with six basic algorithms for CeC2017 test suite.

Algorithm	Indexes	F1	F2	F3	F4	F5
ABC	Mean	6.2080E + 02‡	1.7701E + 11‡	1.2553E + 05‡	**1.3506E + 01**†	9.2203E + 01‡
	Std.	8.0068E + 02	2.9627E + 11	5.0636E + 04	1.7443E + 01	7.8838E + 00
CBO	Mean	1.0451E + 04‡	6.5057E + 13‡	1.5074E + 05‡	5.9317E + 01‡	9.4189E + 01‡
	Std.	2.4186E + 03	1.0556E + 14	6.9549E + 04	3.3481E + 01	1.4968E + 01
FWA	Mean	1.7697E + 10‡	6.5774E + 35‡	7.2439E + 04‡	2.3620E + 03‡	1.8150E + 02‡
	Std.	2.8189E + 09	1.0541E + 36	3.1492E + 03	8.7942E + 02	3.3535E + 01
BSA	Mean	**1.5029E + 03**†	2.6789E + 12‡	3.4571E + 04‡	9.7727E + 01‡	5.4727E + 01‡
	Std.	4.0276E + 02	4.5690E + 12	7.2639E + 03	2.6794E + 01	1.4640E + 01
SCA	Mean	2.2539E + 10‡	4.2264E + 13‡	7.9382E + 06‡	3.5997E + 03‡	3.5153E + 02‡
	Std.	6.1504E + 09	0.0000E + 00	1.1019E + 07	1.5234E + 03	3.5959E + 01
TLBO	Mean	3.6677E + 03‡	9.4050E + 13‡	2.0003E + 03‡	1.0054E + 02‡	1.0521E + 02‡
	Std.	4.5251E + 03	6.6781E + 13	2.2873E + 03	2.4288E + 01	2.1406E + 01
POA	Mean	5.7627E + 02	**0.0000E + 00**	**3.4106E-13**	4.0614E + 01	**4.1457E + 01**
	Std.	6.9515E + 02	0.0000E + 00	1.5039E-13	3.1087E + 01	3.4942E + 00
Rank		2	1	1	2	1
Algorithm	Indexes	F6	F7	F8	F9	F10
ABC	Mean	5.0986E-04†	1.0816E + 02‡	9.3411E + 01‡	9.8807E + 02‡	**2.6599E + 03**†
	Std.	1.7379E-04	1.6494E + 01	1.2142E + 01	4.1642E + 02	1.1324E + 02
CBO	Mean	2.0301E + 00	1.1520E + 02‡	1.1774E + 02‡	2.2280E + 02‡	3.0090E + 03†
	Std.	3.6984E-01	3.3206E + 01	3.7019E + 01	1.9649E + 02	2.9780E + 02
FWA	Mean	4.3271E + 01	3.5782E + 02‡	1.7371E + 02‡	3.1333E + 03‡	4.4572E + 03‡
	Std.	1.0777E + 01	4.0866E + 01	1.2313E + 01	7.7527E + 02	2.7510E + 02
BSA	Mean	**5.9919E-05**†	9.1769E + 01‡	7.1519E + 01‡	**1.7153E-03**†	2.6776E + 03†
	Std.	5.9244E-05	1.0263E + 01	8.8812E + 00	2.5685E-03	6.2866E + 02
SCA	Mean	7.6973E + 01	4.9607E + 02‡	2.7830E + 02‡	8.3535E + 03‡	8.4160E + 03‡
	Std.	2.0904E + 00	4.0586E + 01	2.8177E + 01	2.3462E + 03	1.9877E + 02
TLBO	Mean	1.0878E + 01	1.3238E + 02‡	7.1346E + 01‡	2.7656E + 02‡	6.8244E + 03‡
	Std.	2.2277E + 00	1.9770E + 01	1.4953E + 01	2.5783E + 02	5.5024E + 02
POA	Mean	6.9063E-02	**6.4017E + 01**	**3.8472E + 01**	2.0591E + 00	3.0856E + 03
	Std.	1.1953E-01	3.9269E + 00	6.6247E + 00	2.5473E + 00	5.6373E + 02
Rank		3	1	1	2	4
Algorithm	Indexes	F11	F12	F13	F14	F15
ABC	Mean	6.4484E + 02‡	1.2985E + 06‡	4.4218E + 04‡	2.6949E + 05‡	5.1738E + 03‡
	Std.	2.2023E + 02	5.8008E + 05	2.0387E + 04	5.7495E + 04	2.9482E + 03
CBO	Mean	**4.3705E + 01**†	8.9761E + 05‡	3.5628E + 04‡	9.9343E + 04‡	1.5521E + 03‡
	Std.	7.5790E + 00	9.1348E + 05	5.8350E + 04	9.6147E + 04	1.1989E + 03
FWA	Mean	1.2736E + 03‡	1.1332E + 09‡	9.5208E + 07‡	1.1664E + 06‡	1.0101E + 04‡
	Std.	5.7598E + 02	7.2574E + 08	3.5674E + 07	1.1621E + 06	5.3308E + 03
BSA	Mean	6.0567E + 01†	2.5972E + 05‡	5.4103E + 03‡	**8.3181E + 01**†	**3.0914E + 02**†
	Std.	1.5712E + 01	1.3410E + 05	3.2436E + 03	3.0043E + 01	2.9872E + 02
SCA	Mean	5.6869E + 03‡	4.3194E + 09‡	2.4117E + 09‡	3.4685E + 06‡	9.9570E + 07‡
	Std.	3.5255E + 03	3.5182E + 08	1.3428E + 09	2.6660E + 06	3.0189E + 07
TLBO	Mean	1.7194E + 02‡	5.0899E + 04‡	1.1514E + 04‡	6.9687E + 03‡	4.8375E + 03‡
	Std.	8.2490E + 01	1.0702E + 04	9.1355E + 03	4.1742E + 03	4.8210E + 03
POA	Mean	1.5665E + 02	**7.8858E + 03**	**3.1918E + 03**	2.3114E + 02	3.9830E + 02
	Std.	7.4839E + 01	4.9297E + 03	1.9671E + 03	1.2437E + 02	8.0785E + 01
Rank		3	1	1	2	2
Algorithm	Indexes	F16	F17	F18	F19	F20
ABC	Mean	6.4449E + 02‡	3.1164E + 02‡	3.6940E + 05‡	1.1414E + 04‡	2.5483E + 02‡
	Std.	1.2881E + 02	8.1651E + 01	2.5474E + 04	7.2337E + 03	4.4327E + 01
CBO	Mean	9.0989E + 02‡	4.5551E + 02‡	3.3319E + 05‡	2.0326E + 03‡	2.9056E + 02‡
	Std.	2.7670E + 02	1.1077E + 02	3.0328E + 05	2.4069E + 03	1.1783E + 02
FWA	Mean	1.2949E + 03‡	5.7043E + 02‡	1.4931E + 06‡	2.8732E + 04‡	4.0529E + 02‡
	Std.	3.8149E + 02	1.7929E + 02	7.3002E + 05	3.8499E + 04	1.2227E + 02
BSA	Mean	6.2769E + 02‡	**1.3536E + 02**†	2.9968E + 04‡	**1.0492E + 02**†	2.5063E + 02‡
	Std.	1.0611E + 02	4.0431E + 01	1.2439E + 04	2.5937E + 01	6.5657E + 01
SCA	Mean	2.9177E + 03‡	1.7187E + 03‡	8.4424E + 07‡	8.3600E + 07‡	1.2153E + 03‡
	Std.	5.0769E + 02	3.4788E + 02	6.9263E + 07	7.9817E + 07	4.6527E + 02
TLBO	Mean	6.9030E + 02‡	2.1631E + 02‡	5.9814E + 05‡	3.7965E + 03‡	2.4773E + 02‡
	Std.	2.5551E + 02	3.0783E + 01	3.1696E + 05	3.6412E + 03	1.2790E + 02
POA	Mean	**4.0811E + 02**	1.3988E + 02	**6.0431E + 02**	1.7479E + 02	**1.2172E + 02**
	Std.	3.7920E + 02	1.0166E + 02	2.1307E + 02	1.3400E + 02	1.0740E + 02
Rank		1	2	1	2	1
Algorithm	Indexes	F21	F22	F23	F24	F25
ABC	Mean	2.4867E + 02‡	1.1807E + 02‡	4.2136E + 02‡	4.8982E + 02†	3.8516E + 02†
	Std.	9.8260E + 01	7.5949E + 00	3.6836E + 01	2.4199E + 02	9.0291E-01
CBO	Mean	2.7202E + 02	‡3.7232E + 03‡	4.3592E + 02‡	**4.7082E + 02**†	**3.7895E + 02**†
	Std.	1.7945E + 01	9.5068E + 02	5.6526E + 01	7.8998E + 00	2.9855E + 00
FWA	Mean	3.8718E + 02‡	4.0484E + 03‡	7.7500E + 02‡	1.0242E + 03‡	8.7481E + 02‡
	Std.	6.6261E + 01	9.1657E + 02	6.3272E + 01	1.2852E + 02	4.0566E + 01
BSA	Mean	2.6770E + 02‡	1.0000E + 02‡	4.1554E + 02‡	4.9445E + 02‡	3.8600E + 02~
	Std.	7.7091E + 00	2.7871E-05	1.4893E + 01	2.4847E + 01	1.9619E + 00
SCA	Mean	5.6802E + 02‡	8.7989E + 03‡	8.5778E + 02‡	1.0112E + 03‡	1.2472E + 03‡
	Std.	4.9471E + 01	2.3510E + 02	7.9636E + 01	5.6159E + 01	8.5852E + 01

(*continued on next page*)

Table 4 (*continued*)

Algorithm	Indexes	F1	F2	F3	F4	F5
TLBO	Mean	2.7408E + 02‡	1.0171E + 02‡	4.2967E + 02‡	5.0810E + 02‡	4.0477E + 02‡
	Std.	1.2955E + 01	2.9619E + 00	2.6779E + 01	2.5772E + 01	3.1025E + 01
POA	Mean	**2.3665E + 02**	**1.0000E + 02**	**4.0373E + 02**	4.9297E + 02	3.8608E + 02
	Std.	3.7976E + 00	0.0000E + 00	7.3158E + 00	3.5860E + 01	2.2581E + 00
Rank		1	1	1	3	4
Algorithm	Indexes	F26	F27	F28	F29	F30
ABC	Mean	**2.9936E + 02**†	5.1349E + 02†	4.1879E + 02‡	5.9689E + 02†	1.6954E + 04‡
	Std.	6.0972E + 01	4.7674E + 00	1.8208E + 00	7.3063E + 01	9.4672E + 03
CBO	Mean	1.7964E + 03‡	**5.0001E + 02**†	5.0001E + 02‡	8.0417E + 02†	4.4438E + 03†
	Std.	2.5495E + 02	6.0018E-05	1.1460E-04	2.2036E + 02	5.9191E + 03
FWA	Mean	4.9460E + 03‡	9.6533E + 02‡	1.3642E + 03‡	1.9717E + 03‡	1.5114E + 07‡
	Std.	1.8358E + 03	1.6610E + 02	5.0396E + 02	5.1090E + 02	9.8348E + 06
BSA	Mean	1.1393E + 03†	5.1401E + 02†	4.1111E + 02‡	**4.8204E + 02**†	**4.0515E + 03**†
	Std.	6.8648E + 02	9.6152E + 00	5.1392E + 00	4.6378E + 01	6.1114E + 02
SCA	Mean	4.5519E + 03‡	5.0001E + 02†	5.0001E + 02‡	2.1665E + 03‡	2.2753E + 08‡
	Std.	3.0598E + 02	1.3206E-04	3.6214E-05	5.4765E + 02	2.1657E + 08
TLBO	Mean	1.9164E + 03‡	5.2993E + 02‡	4.3028E + 02‡	8.3072E + 02†	5.2959E + 03†
	Std.	1.4924E + 03	2.0257E + 01	2.5234E + 01	1.9747E + 02	1.0976E + 03
POA	Mean	1.7398E + 03	5.2206E + 02	**3.0000E + 02**	9.8649E + 02	6.9338E + 03
	Std.	2.3361E + 02	1.4190E + 01	2.6357E-13	2.2209E + 02	1.2393E + 03
Rank		3	5	1	5	4

The new individuals are evaluated, and the individuals with the better positions are accepted for next iteration. The algorithm 3 displays the detail steps of this process.

Algorithm 3:

According to their performance, all individuals are sorted in ascending order;
  For $k = N*b + 1{:}N$
    For $d = 1{:}D$
      Execute mutation operator according to equation (11);
      Execute boundary conditions of all variables;
    EndFor
    EvaluateX^* ;
    If X is worse thanX^*, acceptX^*;
  EndFor

3.4. Pseudo-code for POA

The pseudo-code of the whole process of POA can be shown in Algorithm 4.

Algorithm 4

Start
  Initialize N, D,ε, $maxFEs$,a, b, α, β ;
  Initialize current population, history population, store the best individual X_{best};
  While (FEs < maxFEs)
    Renew history population according to equations (9), 10;
    Sort individuals of current and historic population in ascending order;
    Execute **Algorithm 1** to realize sexual propagation;
    Execute **Algorithm 2** to realize asexual propagation;
      Sort individuals of current and historic population in ascending order;
      Execute **Algorithm 3** to realize mutation;
  EndWhile
  Output the final solutions;
End

3.5. Complexity analysis

The complexity of the main factors is analyzed in this part. The complexity of initializing the evolution population and history population is 2.0*$O(N*D)$. To derive the best individual, the complexity of this operator is $O(N)$. The complexity for determining the height of all individuals is 2* $O(N)$. In the process of propagation, the complexity is 1.5*$O(N*D)$. In the algorithm, all individuals are sorted in two times, the complexity of this operator is 2* $O(N^2)$. The approximate complexity of POA is shown in Eq.12. The real complexity of POA is a little larger than the one is derived by the equation, and the mean times which the algorithms reach the acceptable solutions for different functions are listed in the part of experimental results.

$$CA = 2*O\left(N^2\right) + 3.5*O(N*D) + 3*O(N) \tag{12}$$

4. Experimental results

To test the effectiveness of POA, 55 benchmark functions from CEC2005 test suite (Suganthan, Hansen, Liang, Deb & Tiwari, 2005) and CEC2017 test suite (Wu, Mallipeddi & Suganthan, 2017) are tested, and the results are compared with other algorithms with different structures (ABC (Akay & Karaboga, 2012), CBO (Kaveh & Mahdavi, 2014), FWA (Tan and Zhu, 2010), BSA (Civicioglu, 2013), SCA (Mirjalili, 2016), TLBO (Rao et al., 2011), SaDE (Qin, Huang & Suganthan, 2009), jDE (Brest et al., 2006), FIPS (Mendes, Kennedy & Neves, 2004), FDRPSO (Peram, Veeramachaneni & Mohan, 2003), LBSA (Chen, Zou, Lu & Wang, 2017), MBSA (Chen, Zou, Lu & Wang, 2017), ETLBO (Rao & Patel, 2012), DGSTLBO (Zou, Wang, Hei, Chen & Yang, 2014)).

4.1. Parameter settings

To obtain a fair comparison and reduce the statistical errors, all algorithms were independently executed 50 times independently, and the results of 50 independent runs on all functions are statistically analyzed. All algorithms have the same training parameters. The population size is 50, and the maximal function evaluation times (FEs) is 5000*D, where D is the dimension of the decision variables. Some specific parameters of algorithms are taken from the associated references.

Our proposed algorithm requires values for the parameter a for asexual reproduction, the parameter b for mutation operator, the height factorα, and the updating coefficientβ . In the experiments, we selected a from the set {0.2, 0.4, …, 0.8, 1.0} while$b = 0.2,\ \alpha = 0.6,\ \beta = 1.4$, b from the set {0.2, 0.4, …, 0.8, 1.0} while$a = 0.6,\ \alpha = 0.6,\ \beta = 1.4$, α from the set {0.2, 0.4, …, 0.8, 1.0} while $a = 0.6,\ b = 0.2,\ \beta = 1.4$ and β from the set {0.8,1.0, 1.2, 1.4, 1.6} while$a = 0.6,\ b = 0.2,\ \alpha = 0.6$. Then POA with the different $a,\ b,\ \alpha,\ \beta$ is performed for function F2, F4, F8, F12, F16, F21 and F24 from CEC2017 test suite and then determined the most appropriate values. The results are given in Table 1.

It can be observed from Table 1 that POA can find the theoretical optimal solution with $a = 0.2$, 0.6 and 0.8 for F2. POA performed significantly better with $a = 0.2$ for F4, $a = 0.4$ for F8, $a = 0.6$ for F16 and F24, and $a = 0.8$ for F12 and F21. Based on the above results, we set $a = 0.6$ to attain a balance between solution accuracy for simple and complex problems. It can also be observed from Table 1 that POA can find the theoretical optimal solution with $b = 0.6$ and 0.8 for F2. POA

Table 5
Comparison of POA with eight variants of basic algorithms for CEC2017 test suite.

Algorithm	Indexes	F1	F2	F3	F4	F5
jDE	Mean	**3.3159E-14**†	5.1633E + 06‡	5.4212E-01‡	**1.4824E + 01**†	**4.1275E + 01**†
	Std.	2.1707E-14	6.9053E + 06	3.3933E-01	2.7034E + 00	2.1565E + 00
SaDE	Mean	5.7502E-07†	5.0000E + 00‡	1.6984E + 01‡	2.4904E + 01†	7.7898E + 01‡
	Std.	9.9588E-07	8.6603E + 00	1.7477E + 01	3.9730E + 01	1.9962E + 00
PSOwFIPS	Mean	1.0505E + 04‡	1.0812E + 16‡	1.9620E + 04‡	6.2127E + 01‡	1.4980E + 02‡
	Std.	7.4963E + 03	1.0760E + 16	3.0969E + 03	2.9677E + 01	1.9493E + 01
FDRPSO	Mean	7.0714E + 03‡	6.3397E + 16‡	3.2340E + 02‡	9.4243E + 01‡	6.4120E + 01‡
	Std.	2.9249E + 03	8.1559E + 16	1.5160E + 02	2.6679E + 01	2.6536E + 01
LBSA	Mean	3.0324E-10†	4.3189E + 04‡	1.3886E + 03‡	4.5524E + 01‡	5.5621E + 01‡
	Std.	1.5608E-10	7.3622E + 04	7.0714E + 02	3.6424E + 01	9.7588E + 00
MBSA	Mean	2.3685E-14†	2.1444E + 11‡	1.5119E-02‡	4.6569E + 01‡	5.3417E + 01‡
	Std.	8.2046E-15	3.6989E + 11	1.3173E-02	4.0084E + 01	8.4871E + 00
ETLBO	Mean	3.3073E + 03‡	9.4897E + 10‡	1.9161E + 03‡	9.0384E + 01‡	1.0124E + 02‡
	Std.	3.0876E + 03	1.6334E + 11	1.9833E + 03	2.3267E + 01	1.3814E + 01
DGSTLBO	Mean	7.9603E + 03‡	7.7445E + 23‡	4.3997E-03‡	1.2328E + 02‡	8.4870E + 01‡
	Std.	4.0342E + 03	2.2714E + 23	4.0081E-03	1.1692E + 01	8.0922E + 00
POA	Mean	5.7627E + 02	**0.0000E + 00**	**3.4106E-13**	4.0614E + 01	4.1457E + 01
	Std.	6.9515E + 02	0.0000E + 00	1.5039E-13	3.1087E + 01	3.4942E + 00
Rank		5	1	1	3	2
Algorithm	Indexes	F6	F7	F8	F9	F10
jDE	Mean	**1.1369E-13**†	7.0982E + 01‡	5.2897E + 01‡	**3.7896E-14**†	3.2718E + 03‡
	Std.	0.0000E + 00	6.9609E + 00	1.0081E + 01	6.5637E-14	5.0175E + 02
SaDE	Mean	**1.1369E-13**†	1.2529E + 02‡	7.3633E + 01‡	2.9843E-02†	4.3924E + 03‡
	Std.	0.0000E + 00	1.2209E + 01	7.0716E + 00	5.1689E-02	3.0286E + 02
PSOwFIPS	Mean	3.9573E-03†	2.1567E + 02‡	1.5745E + 02‡	1.9714E-03†	6.7816E + 03‡
	Std.	5.8093E-04	7.6769E + 00	7.7335E + 00	1.4058E-03	1.0206E + 02
FDRPSO	Mean	6.6578E-01‡	1.0118E + 02‡	7.2643E + 01‡	2.8519E + 01‡	3.3255E + 03‡
	Std.	5.3463E-01	1.0487E + 01	2.0828E + 01	2.5251E + 01	5.2891E + 02
LBSA	Mean	1.6909E-04†	7.3444E + 01‡	4.9314E + 01‡	1.0899E + 00†	**2.6674E + 03**†
	Std.	2.7669E-04	2.6985E + 00	1.2673E + 01	1.4285E + 00	3.1350E + 02
MBSA	Mean	8.3615E-02‡	1.0290E + 02‡	5.2939E + 01‡	2.6604E + 00‡	3.3689E + 03‡
	Std.	1.1872E-01	1.1123E + 01	9.2996E + 00	3.0520E + 00	5.2673E + 02
ETLBO	Mean	7.0688E + 00‡	1.1975E + 02‡	8.2581E + 01‡	3.2305E + 02‡	6.8557E + 03‡
	Std.	2.9753E + 00	1.3977E + 01	9.9491E-01	3.6661E + 02	3.4423E + 02
DGSTLBO	Mean	1.5933E + 01‡	1.9823E + 02‡	8.7510E + 01‡	3.7860E + 02‡	2.8867E + 03†
	Std.	4.9187E + 00	3.7696E + 01	2.2400E + 01	1.2526E + 02	8.2069E + 02
POA	Mean	6.9063E-02	**6.4017E + 01**	**3.8472E + 01**	2.0591E + 00	3.0856E + 03
	Std.	1.1953E-01	3.9269E + 00	6.6247E + 00	2.5473E + 00	5.6373E + 02
Rank		5	1	1	5	3
Algorithm	Indexes	F11	F12	F13	F14	F15
jDE	Mean	**1.8941E + 01**†	4.4833E + 04‡	1.0540E + 04‡	**3.3665E + 01**†	4.8528E + 01†
	Std.	6.2281E + 00	1.7420E + 04	1.2483E + 04	5.8922E + 00	4.4316E + 01
SaDE	Mean	3.8544E + 01†	1.0634E + 04‡	**2.1452E + 03**†	3.6402E + 01†	**2.4206E + 01**†
	Std.	2.7388E + 01	2.6044E + 03	1.5360E + 03	1.8196E + 01	1.0793E + 01
PSOwFIPS	Mean	6.0415E + 01†	1.5070E + 06‡	4.0365E + 04‡	9.9150E + 03‡	5.9792E + 03‡
	Std.	1.4065E + 01	9.1548E + 05	5.4414E + 03	7.6781E + 03	4.5463E + 03
FDRPSO	Mean	1.7636E + 02‡	3.6618E + 05‡	2.3989E + 04‡	6.3056E + 03‡	3.1427E + 03‡
	Std.	2.0708E + 01	4.5900E + 05	2.0280E + 04	6.9181E + 03	3.3481E + 03
LBSA	Mean	4.0378E + 01†	2.5731E + 04‡	1.4736E + 04‡	6.4500E + 01†	4.7580E + 02‡
	Std.	1.6501E + 01	5.8328E + 03	5.6410E + 03	1.5594E + 01	6.4116E + 02
MBSA	Mean	9.1303E + 01†	1.1580E + 04‡	4.4001E + 03‡	7.1662E + 01†	1.1235E + 03‡
	Std.	3.4282E + 01	6.9082E + 03	2.8784E + 03	1.9436E + 01	1.1565E + 03
ETLBO	Mean	7.0666E + 01†	2.6025E + 04‡	2.9405E + 04‡	6.6746E + 03‡	1.6494E + 03‡
	Std.	1.8639E + 01	1.3200E + 04	2.8288E + 04	6.9656E + 03	1.4278E + 03
DGSTLBO	Mean	1.1663E + 02†	8.4770E + 05‡	1.9072E + 04‡	5.0760E + 03‡	3.5208E + 03‡
	Std.	3.2487E + 01	9.1545E + 05	1.2338E + 04	3.6161E + 03	3.0048E + 03
POA	Mean	1.5665E + 02	**7.8858E + 03**	3.1918E + 03	2.3114E + 02	3.9830E + 02
	Std.	7.4839E + 01	4.9297E + 03	1.9671E + 03	1.2437E + 02	8.0785E + 01
Rank		8	1	2	5	3
Algorithm	Indexes	F16	F17	F18	F19	F20
jDE	Mean	8.1577E + 02‡	2.3577E + 02‡	8.0334E + 03‡	**1.7777E + 01**†	1.8715E + 02‡
	Std.	9.5829E + 01	7.9017E + 01	1.0383E + 04	4.0740E + 00	1.7578E + 01
SaDE	Mean	4.7814E + 02‡	**4.9875E + 01**†	2.4232E + 03‡	1.9419E + 01†	1.5835E + 02‡
	Std.	1.5861E + 02	1.2455E + 01	3.4301E + 03	5.4144E + 00	6.4103E + 00
PSOwFIPS	Mean	1.1273E + 03‡	2.2532E + 02‡	4.0915E + 05‡	2.1932E + 03‡	2.4398E + 02‡
	Std.	2.7154E + 02	9.6255E + 01	1.2570E + 05	1.4873E + 03	3.0933E + 01
FDRPSO	Mean	8.1979E + 02‡	3.7510E + 02‡	5.9172E + 04‡	8.4658E + 03‡	3.4471E + 02‡
	Std.	1.4599E + 02	1.7647E + 02	3.3985E + 04	1.1218E + 04	1.2122E + 02
LBSA	Mean	**2.7523E + 02**†	8.0681E + 01†	2.3032E + 04‡	4.2607E + 01†	1.2910E + 02‡
	Std.	7.4243E + 01	7.7531E + 01	1.2212E + 04	2.5160E + 01	6.7923E + 01
MBSA	Mean	5.5728E + 02‡	1.7096E + 02‡	6.1268E + 03‡	3.8767E + 02‡	1.5272E + 02‡
	Std.	2.7003E + 02	1.0711E + 02	3.3695E + 03	5.3463E + 02	4.6087E + 00
ETLBO	Mean	6.7350E + 02‡	2.3053E + 02‡	3.6623E + 05‡	4.8366E + 03‡	2.9995E + 02‡
	Std.	8.9387E + 01	1.8242E + 02	7.3220E + 04	6.3799E + 03	5.5594E + 01

(*continued on next page*)

Table 5 (*continued*)

Algorithm	Indexes	F1	F2	F3	F4	F5
DGSTLBO	Mean	6.5924E + 02‡	2.8189E + 02‡	1.2964E + 05‡	1.0024E + 04‡	3.2014E + 02‡
	Std.	3.4572E + 02	5.0541E + 01	6.8064E + 04	8.6420E + 03	5.1853E + 01
POA	Mean	4.0811E + 02	1.3988E + 02	**6.0431E + 02**	1.7479E + 02	**1.2172E + 02**
	Std.	3.7920E + 02	1.0166E + 02	2.1307E + 02	1.3400E + 02	1.0740E + 02
Rank		2	3	1	4	1
Algorithm	Indexes	F21	F22	F23	F24	F25
jDE	Mean	2.6671E + 02‡	1.1902E + 03‡	3.9309E + 02†	4.7676E + 02†	3.7838E + 02†
	Std.	1.2306E + 01	1.8884E + 03	2.7170E + 00	1.1103E + 01	1.7791E-01
SaDE	Mean	2.5930E + 02‡	**1.0000E + 02**~	3.9526E + 02†	**4.5401E + 02**†	3.8699E + 02‡
	Std.	3.3454E + 01	0.0000E + 00	1.1251E + 01	1.1229E + 01	1.6128E-01
PSOwFIPS	Mean	3.2989E + 02‡	1.0001E + 02~	5.1278E + 02‡	6.0255E + 02‡	**3.7843E + 02**†
	Std.	1.2585E + 01	1.2163E-03	5.9126E + 00	2.2446E + 01	1.0996E-01
FDRPSO	Mean	2.7097E + 02‡	1.3580E + 03‡	4.7613E + 02‡	5.2747E + 02‡	3.8797E + 02‡
	Std.	1.2581E + 01	1.6239E + 03	7.9338E + 01	1.2475E + 01	9.4451E-01
LBSA	Mean	2.5128E + 02‡	1.0115E + 02‡	4.0393E + 02~	4.6816E + 02†	3.8759E + 02‡
	Std.	1.3753E + 01	1.9925E + 00	1.0844E + 01	5.9022E + 00	1.6979E-01
MBSA	Mean	2.3906E + 02‡	1.0126E + 02‡	**3.9178E + 02**†	4.6990E + 02†	3.8662E + 02‡
	Std.	9.8676E + 00	2.1823E + 00	2.2637E + 01	5.7980E + 00	2.5964E + 00
ETLBO	Mean	2.5739E + 02‡	1.0161E + 02‡	4.7136E + 02‡	4.9748E + 02‡	4.0809E + 02‡
	Std.	6.8032E + 00	2.7831E + 00	4.2218E + 01	2.3306E + 01	2.4733E + 01
DGSTLBO	Mean	2.8516E + 02‡	1.1047E + 02‡	4.9967E + 02‡	5.5201E + 02‡	4.3145E + 02‡
	Std.	8.2099E + 00	4.1367E + 00	2.2495E + 01	8.0517E + 00	2.7443E + 01
POA	Mean	**2.3665E + 02**	**1.0000E + 02**	4.0373E + 02	4.9297E + 02	3.8608E + 02
	Std.	3.7976E + 00	0.0000E + 00	7.3158E + 00	3.5860E + 01	2.2581E + 00
Rank		1	1	4	5	3
Algorithm	Indexes	F26	F27	F28	F29	F30
jDE	Mean	1.4235E + 03†	5.0001E + 02†	4.9804E + 02‡	4.8428E + 02†	**2.1866E + 02**†
	Std.	2.1947E + 02	1.1059E-04	3.4012E + 00	9.0465E + 01	6.3004E + 00
SaDE	Mean	1.3834E + 03†	5.0486E + 02†	3.9428E + 02‡	5.6364E + 02†	3.7908E + 03†
	Std.	1.3766E + 02	4.4457E + 00	6.8317E-01	3.6053E + 01	7.8411E + 02
PSOwFIPS	Mean	2.7290E + 02†	**4.4242E + 02**†	4.1270E + 02‡	6.9825E + 02†	4.1347E + 04‡
	Std.	6.3600E + 01	1.8930E + 00	3.1729E + 00	5.8227E + 01	3.0062E + 04
FDRPSO	Mean	**1.3647E + 03**†	5.7040E + 02‡	4.3360E + 02‡	7.1438E + 02†	7.3533E + 04‡
	Std.	1.0582E + 03	6.0670E + 01	2.1687E + 01	1.9316E + 02	5.6009E + 04
LBSA	Mean	1.4774E + 03†	5.0909E + 02†	3.3571E + 02‡	**4.8206E + 02**†	2.7815E + 03†
	Std.	8.9418E + 01	5.7485E + 00	6.1858E + 01	8.8572E + 00	6.7272E + 02
MBSA	Mean	1.5514E + 03†	5.4381E + 02‡	3.6126E + 02‡	5.4986E + 02†	3.3647E + 03†
	Std.	1.5619E + 02	1.9885E + 01	5.3057E + 01	4.7078E + 01	1.1396E + 03
ETLBO	Mean	2.8195E + 03‡	5.5361E + 02‡	3.7278E + 02‡	9.2721E + 02†	5.0752E + 03†
	Std.	1.0878E + 03	3.9961E + 01	6.3501E + 01	1.9551E + 02	4.4034E + 03
DGSTLBO	Mean	1.9791E + 03‡	6.0693E + 02‡	4.6501E + 02‡	9.0956E + 02†	6.8032E + 03†
	Std.	1.0024E + 03	7.0533E + 01	1.3779E + 01	5.6701E + 01	2.7246E + 03
POA	Mean	1.7398E + 03	5.2206E + 02	**3.0000E + 02**	9.8649E + 02	6.9338E + 03
	Std.	2.3361E + 02	1.4190E + 01	2.6357E-13	2.2209E + 02	1.2393E + 03
Rank		7	5	1	9	7

performed significantly better with $b = 0.2$ for F8, F16 and F24, $b = 0.4$ for F4 and F21, and $b = 0.6$ for F12. Based on the above results, we set $b = 0.2$ to attain a balance between solution accuracy for simple and complex problems. It can be seen from Table 1 that POA can find the theoretical optimal solution with$\alpha = 0.4$ for F2. POA performed significantly better with$\alpha = 0.2$ for F4 and F24, α=0.4 for F2 and F16, α=0.6 for F8 and F12, and$\alpha = 0.8$ for F21. Based on the above results, we set$\alpha = 0.6$ to attain a balance between solution accuracy for simple and complex problems. It can also be seen from Table 1 that POA can find the theoretical optimal solution with$\beta = 0.8$, 1.4 and 1.6 for F2. POA performed significantly better with$\beta = 0.8$, 1.4 and 1.6 for F2, β=1.0 for F4 and F12, β=1.4 for F8, F21 and F24, and$\beta = 1.6$ for F12. Based on the above results, we set$\beta = 1.4$ to attain a balance between solution accuracy for simple and complex problems.

Based on the above analysis, we set $a = 0.6,\ b = 0.2,\ \alpha = 0.6,\ \beta = 1.4$ to attain a balance between solution accuracy for CEC2005 test suite and cec2017 test suite in the following experiments.

4.2. Experiments for functions of CEC2005 test suite

In this sub-experiment, we compare 14 algorithms including POA on 25 functions from CEC2005 test suite (Suganthan et al., 2005). In the experiments, the fitness function is the absolute Error between theoretical and real optimal values. If the theoretical solution of a function isy^*, and the real solution is y, the fitness function is$F = |y^* - y|$. For all functions, the variable dimension is 30 (D = 30), the statistical results of the obtained solutions are given in the follow tables. Table 2 gives the comparison results between POA and some basic optimization algorithms and Table 3 gives the comparison results between POA and some variants of basic optimization algorithms. Need to note that, "Rank" in Table 2 and Table 3 represents the competitive ranks of all algorithms for each function. The best solutions are marked by the boldface. To analyze performances of algorithms, a *t*-test (Zhan, Zhang, Li & Chung, 2009) further statistically with a significance level of 0.05 are performed and the results are also given in the two tables. The '‡', '~'and †' respectively refer to the number that POA is significantly better than, similarly to and worse than other algorithms.

In term of comparison of POA with six basic algorithms, Table 2 shows that POA performs well for 12 out of 25 functions among all seven algorithms. POA ranked the second competitively with other algorithms for 3 out of 25 functions and the third competitively with other algorithms for 8 out of 25 functions. For functions F9, F13, F15, F21 and F23, ABC obtained the best results in the 50 runs. For functions F18, F19, F20, F22 and F25, CBO are better than all other algorithms in term of the obtained mean solutions. BSA performs well for the functions F16 and F24, and TLBO performs well for the function F12. Expressly, although POA is not better than some other algorithm in term of the obtained mean solutions, the competitive ranks of POA are all in the top three for

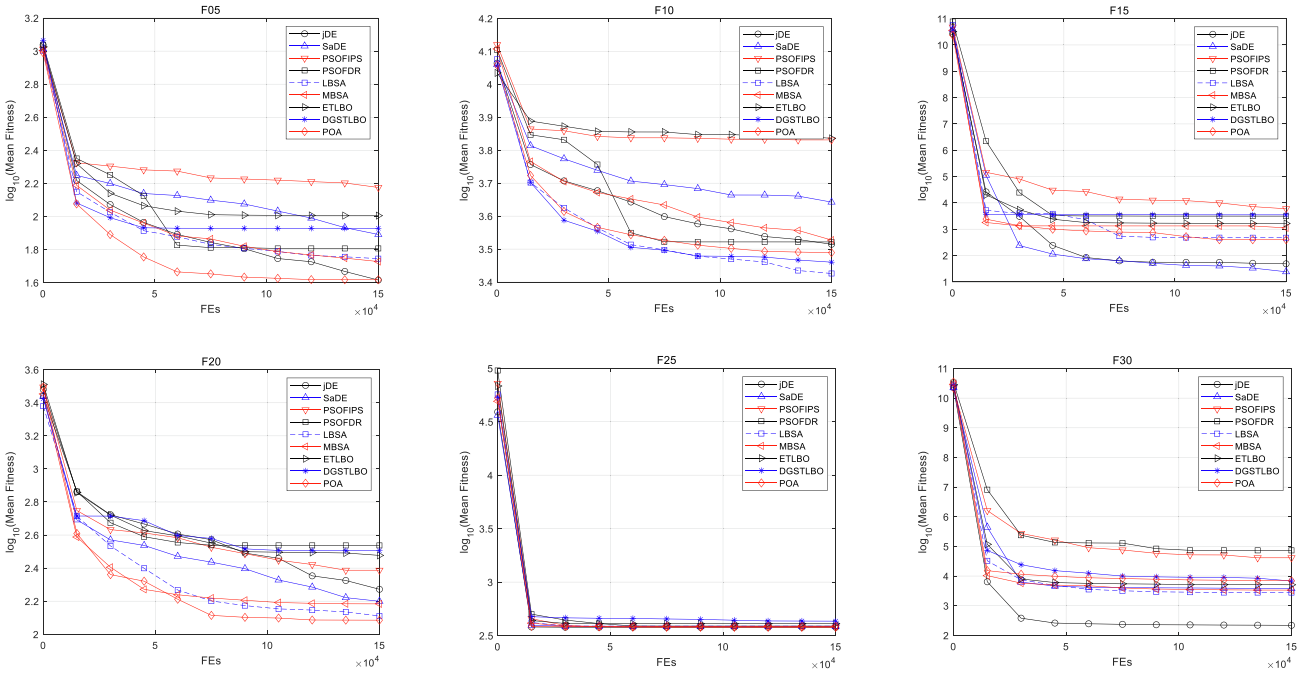

Fig. 5. Comparison of the performance curves using different algorithms.

Table 6
Optimal threshold values and metrics obtained by POA.

Image	K	Threshold	PSNR	SSIM	FSIM
Lena	2	98 165	18.8976	0.97496	0.81843
	4	77 114 149 184	23.4757	0.98915	0.90481
	6	62 90 117 142 168 197	25.5185	0.99269	0.9344
	8	59 80 100 122 143 163 183 204	27.1854	0.99467	0.95498
Baboon	2	80 144	19.4716	0.98117	0.88485
	4	34 75 115 160	21.5193	0.98325	0.93598
	6	29 56 85 114 145 176	23.7596	0.98838	0.96077
	8	18 39 63 88 114 140 166 193	25.2813	0.99064	0.98008
Starfish	2	92 170	19.2433	0.97478	0.8197
	4	68 116 163 206	21.2304	0.97796	0.86623
	6	50 81 114 147 179 214	24.6867	0.98483	0.9299
	8	43 68 92 117 142 169 195 222	25.5330	0.98956	0.95504
Butterfly	2	116 175	26.1402	0.99906	0.95473
	4	74 119 172 220	21.4242	0.98377	0.89776
	6	62 90 119 150 180 221	24.9638	0.99130	0.93157
	8	50 73 98 124 151 178 204 228	25.7199	0.99235	0.94759
T1down256	2	115 167	41.3986	0.99996	0.99736
	4	28 71 115 167	26.2832	0.89950	0.89571
	6	22 54 84 115 150 186	28.1964	0.96087	0.93792
	8	20 44 68 91 115 141 170 199	30.1797	0.96869	0.95811
Skull	2	130 182	26.7890	0.99611	0.92630
	4	34 84 133 183	22.2310	0.96765	0.90103
	6	30 67 102 139 178 216	24.6993	0.98139	0.94086
	8	9 45 78 110 142 176 202 229	25.1872	0.98192	0.95142

all functions except for functions F18 and F24. From the results (‡/∼/ †) in Table 2, it can be concluded that POA performed significantly better than other six basic algorithms. For example, POA performs better for 19 functions and worse for 6 functions from 25 functions with 30-D decision variables compared with ABC. These indicate clearly that POA overall outperforms ABC.

In term of comparison of POA with eight variants of basic algorithms, Table 3 shows that POA performs well for 10 out of 25 functions among all algorithms. POA ranked the second competitively with other algorithms for 2 out of 25 functions and the third competitively with other algorithms for 4 out of 25 functions. For functions F1 and F9, jDE, SaDE, LBSA and MBSA can all find the theoretical optimal values. For 6 functions F13, F15, F18, F19, F20 and F22, jDE obtained the best results in the 50 runs. SaDE performs well for 3 functions F7, F21 and F23, PSOwFIPS performs well for function F25, FDRPSO performs well for function F14, and MBSA performs well for 3 functions F12, F16 and F24. Expressly, the competitive ranks of POA are all in the top three for 17 out of 25 functions compared with other algorithms in term of the obtained mean solutions. Besides, it can be observed the results (‡/∼/ †) in Table 3 that, POA performs better than other eight variants of basic algorithms for the most of all CEC2005 test functions with 30-D variables. Fig. 4 presents representative convergence graphs of the average fitness values for some typical functions.

4.3. Experiments for functions of CEC2017 test suite

In this sub-experiment, we compare 14 algorithms including POA on 30 functions from CEC2017 test suite (Wu et al., 2017). For all functions, the variable dimension is 30 (D = 30), the statistical results of the obtained solutions are given in the follow tables. The boldface indicates that the solution is the best one. To further statistically analyze performances of algorithms, a *t*-test [27] with a significance level of 0.05 are performed and the results are given in the follow two tables. The '‡', '∼'and †' respectively refer to the number that POA is significantly better than, similarly to and worse than other algorithms。.

In term of comparison of POA with six basic algorithms, Table 4 indicates that POA performs well for 14 out of 30 functions among all seven algorithms. POA ranked the second competitively with other algorithms for 7 functions F1, F4, F9, F14, F15, F17, and F19, and the third competitively with other algorithms for 4 functions F6, F11, F24 and F26. When solving 2 functions F4 and F10, ABC obtained the best results in the 50 runs. CBO performs well for 4 functions F11, F24, F25 and F27. For 9 functions F1, F6, F9, F14, F15, F17, F19, F29 and F30, BSA obtained the best results in the 50 runs. Expressly, although POA is not better than some other algorithm in term of the obtained mean solutions, the competitive ranks of POA are all in the top three for 25 out of 30

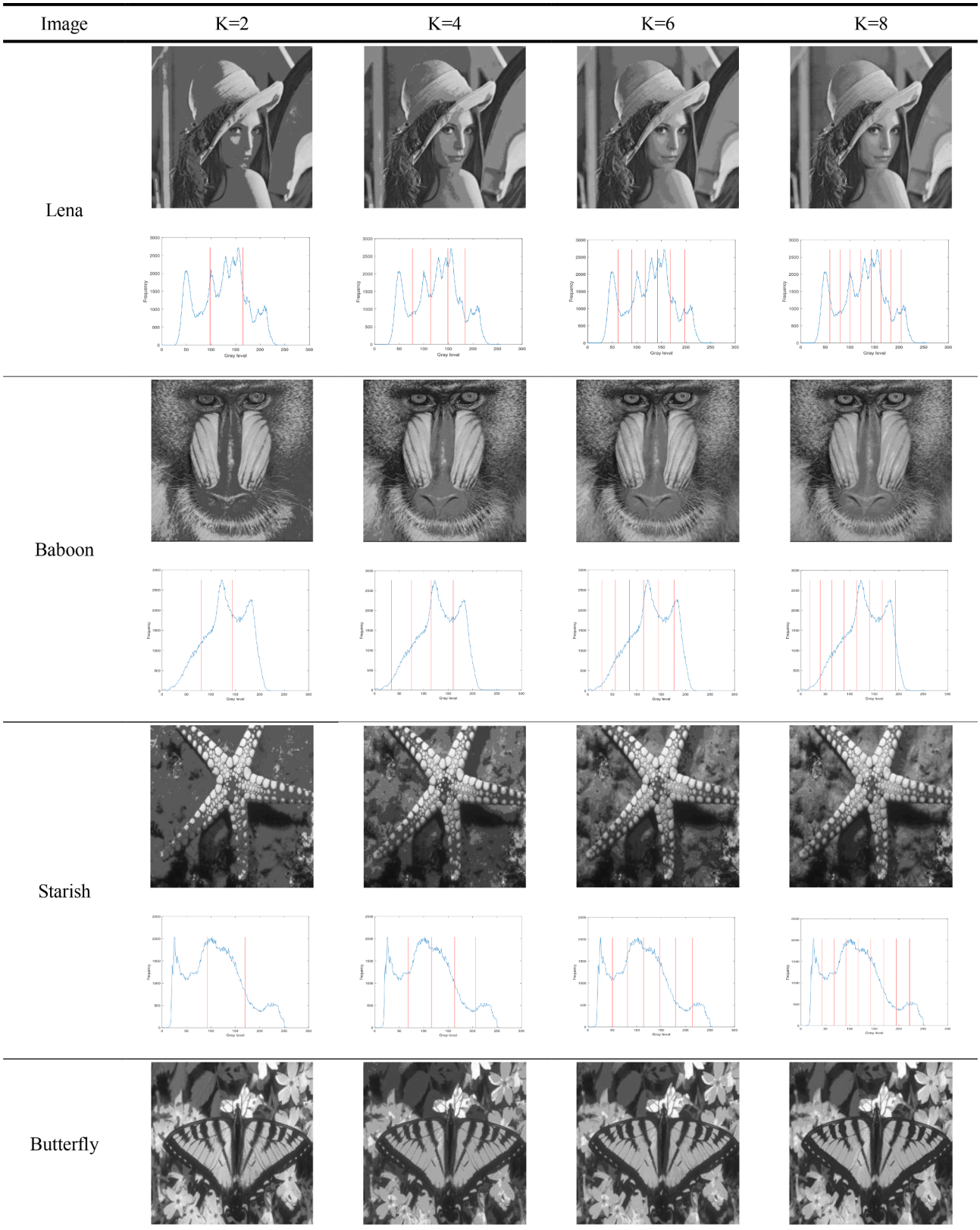

Fig. 6. Results with different threshold levels on six images obtained by POA.

functions except for function F10, F25, F27, F29 and F30 compared with other algorithms. From the results (‡/∼/ †) in Table 4, it displays that POA performed significantly better than other six basic algorithms and obtained a good optimization performance. For example, POA performs better for 21 functions and worse for 8 functions from 30 functions with 30-D decision variables compared with ABC. Expressly, POA performed better than FWA for all 30 functions. These indicate clearly that POA overall outperforms ABC and FWA.

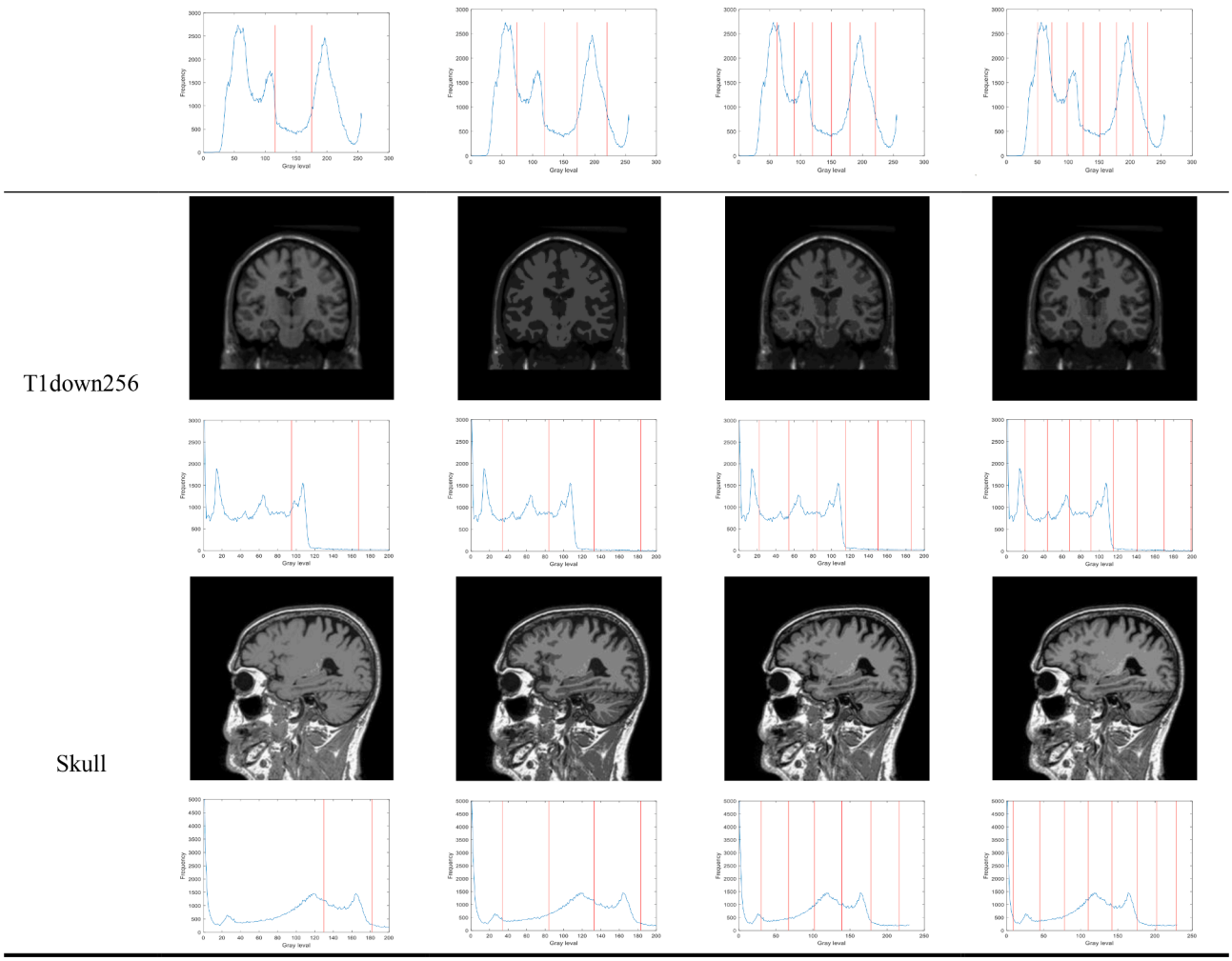

Fig. 6. (*continued*).

Table 7
Comparison of PSNR values obtained by different algorithms.

Image	K	FDRPSO	jDE	SaDE	POA
Lena	2	**18.8976**	18.7521	**18.8976**	**18.8976**
	4	22.6053	20.7031	22.4481	**22.6323**
	6	24.9191	20.8588	25.4424	**25.5143**
	8	26.5099	18.8938	26.577	**26.8717**
Baboon	2	**19.4716**	19.0666	**19.4716**	**19.4716**
	4	21.4658	20.0211	**21.5193**	**21.5193**
	6	22.8946	18.2373	23.1452	**23.6359**
	8	24.7288	19.3426	24.8987	**24.9685**
Starfish	2	19.2252	18.8042	**19.2433**	**19.2433**
	4	**21.3283**	18.9455	21.2304	21.2304
	6	23.6136	19.2282	24.3282	**24.5167**
	8	25.2522	20.5442	25.3335	**25.3567**
Butterfly	2	**26.1402**	22.2251	**26.1402**	**26.1402**
	4	**21.4253**	17.8115	21.4242	21.4242
	6	24.0680	18.1342	24.0432	**24.2631**
	8	25.3414	19.1044	25.5447	**25.5769**
T1down256	2	**41.3986**	40.6651	**41.3986**	**41.3986**
	4	25.947	21.6709	25.0205	**26.0368**
	6	27.5746	21.6709	**28.1874**	28.1067
	8	29.4584	27.9527	**30.0424**	29.6714
Skull	2	**26.7890**	23.5847	**26.7890**	**26.7890**
	4	21.7827	18.2255	**22.2310**	**22.2310**
	6	23.5006	18.2255	23.1494	**23.6623**
	8	**25.5424**	21.6468	25.1252	25.0906

Table 8
Comparison of SSIM values obtained by different algorithms.

Image	K	FDRPSO	jDE	SaDE	POA
Lena	2	**0.97496**	0.9696	**0.97496**	**0.97496**
	4	0.98626	0.97350	0.98575	**0.98636**
	6	0.99101	0.97740	0.99236	**0.99248**
	8	0.99347	0.95222	**0.99412**	0.9936
Baboon	2	**0.98117**	0.97886	**0.98117**	**0.98117**
	4	0.98289	**0.98460**	0.98325	0.98325
	6	0.98371	0.92920	0.98773	**0.98832**
	8	0.98916	0.9615	0.98971	**0.98986**
Starfish	2	0.97462	0.97059	**0.97478**	**0.97478**
	4	0.97662	0.9680	0.97596	**0.97796**
	6	0.98456	0.94040	**0.98909**	0.98417
	8	0.9889	0.93943	0.98911	**0.98914**
Butterfly	2	**0.99906**	0.99631	**0.99906**	**0.99906**
	4	**0.98377**	0.93310	**0.98377**	**0.98377**
	6	0.99141	0.94030	0.99028	**0.99152**
	8	0.99133	0.96209	0.99191	**0.99193**
T1down256	2	**0.99996**	0.99995	**0.99996**	**0.99996**
	4	**0.92476**	0.65750	0.89418	0.89471
	6	0.94013	0.92880	0.95034	**0.95974**
	8	0.96123	0.92547	0.96743	**0.96812**
Skull	2	**0.99611**	0.87453	**0.99611**	**0.99611**
	4	0.96336	0.92880	**0.96765**	**0.96765**
	6	0.97343	0.83840	0.97048	**0.97483**
	8	**0.98338**	0.95899	0.98158	0.98144

Table 9
Comparison of FSIM values obtained by different algorithms.

Image	K	FDRPSO	jDE	SaDE	POA
Lena	2	**0.81843**	0.80615	**0.81843**	**0.81843**
	4	**0.88833**	0.86190	0.88321	0.88752
	6	0.92442	0.85580	**0.93433**	0.93243
	8	0.94653	0.81383	0.94078	**0.94755**
Baboon	2	**0.88485**	0.87800	**0.88485**	**0.88485**
	4	0.93517	0.90140	**0.93598**	**0.93598**
	6	0.95903	0.86400	**0.95993**	0.95707
	8	0.97597	0.88590	0.97526	**0.97686**
Starfish	2	0.81925	0.80838	**0.81970**	**0.81970**
	4	**0.86837**	0.82340	0.86623	0.86623
	6	0.92134	0.81370	0.91785	**0.92959**
	8	0.94658	0.83085	0.94604	**0.94738**
Butterfly	2	**0.95473**	0.90805	**0.95473**	**0.95473**
	4	**0.89781**	0.79440	0.89776	0.89776
	6	**0.93203**	0.82450	0.92644	0.92761
	8	0.94371	0.84287	0.94623	**0.94638**
T1down256	2	**0.99736**	0.99675	**0.99736**	**0.99736**
	4	**0.90370**	0.83840	0.89251	0.89283
	6	0.92776	0.85130	0.93196	**0.93431**
	8	0.95009	0.93246	**0.95563**	0.95202
Skull	2	**0.92630**	0.82494	**0.92630**	**0.92630**
	4	0.88944	0.78630	**0.90103**	**0.90103**
	6	**0.93012**	0.84490	0.92547	0.92881
	8	**0.95481**	0.87084	0.95100	0.95064

In term of comparison of POA with eight variants of basic algorithms, Table 5 indicates that POA performs well for 10 out of 30 functions among all seven algorithms. POA ranked the second competitively with other algorithms for 3 functions F5, F13, and F16, and the third competitively with other algorithms for 5 functions F4, F10, F15, F17 and F25. When solving the functions F6, jDE and SaDE can all perform well. For 8 functions F1, F4, F5, F9, F11, F14, F19 and F30, jDE obtained the best results in the 50 runs. SaDE performs well for 5 function F13, F15, F17, F22 and F24, PSOwFIPS performs well for 2 functions F25 and F27, LBSA performs well for 3 functions F10, F16 and F29, and MBSA performs well for function F23. Expressly, although POA is not better than some other algorithm in term of the obtained mean solutions, the competitive ranks of POA are all in the top three for 18 out of 30 functions compared with other algorithms. Besides, it can be observed the results (‡/∼/ †) in Table 5 that, POA don't perform significantly better than other eight variants of basic algorithms. For example, POA performs better for 14 functions and worse for 16 functions from 30 functions with 30-D decision variables compared with jDE. These indicate clearly that POA overall don't outperforms jDE. Hence, our algorithm needs further improvement. Fig. 5 presents representative convergence graphs of the average fitness values for some typical functions.

5. Multilevel thresholding image segmentation

Image segmentation can be considered as an optimization problem of obtaining optimal thresholds according to their characteristics for image segmentation. In this section, six different images (Lena, Baboon, Cameraman, Starfish, Butterfly, T1down256 and Skull) are used to test our proposed approach and they are widely used to test different segment methods. To compare the segment performance, PSNR (Wang, Bovik, Sheikh & Simoncelli, 2004), SSIM (Agrawal, Panda, Bhuyan & Panigrahi, 2013) and FSIM (He and Huang, 2020) are used in the paper.

Table 6 gives the optimal threshold values and metrics obtained by our proposed POA method at one run for benchmark images with different segmentation levels 2, 4, 6 and 8. Moreover, Fig. 6 respectively shows the segmentation results with different threshold levels of different benchmark test images by our proposed POA method.

It can also be seen from Fig. 6 that the segmented images provide evidence that the outcome is better with the increasement of the number of the threshold, and they are consistent with the conclusion in terms of PSNR, SSIM and FSIM in Table 6. Furthermore, the PSNR, SSIM and FSIM values of different algorithms over benchmark images are respectively given in Tables 7-9.

It can be seen from Table 7 that, POA has the highest accuracy performance on all images with four different segmentation levels 2, 4, 6 and 8 in terms of the best values of PSNR. FDRPSO performs well on Starfish and Butterfly with the segmentation level 4 and Skull with the segmentation level 8. SaDE performs well on T1down256 with the segmentation levels 6 and 8. POA performs well in the other cases.

It can be seen from Table 8 that, POA has the highest accuracy performance on all images with four different segmentation levels 2, 4, 6 and 8 in terms of the best values of SSIM. FDRPSO performs well on T1down256 with the segmentation level 4 and Skull with the segmentation level 8. jDE performs well on Baboon with the segmentation level 4. SaDE performs well on Lena with the segmentation level 8 and Starfish with the segmentation level 6. POA performs well in the other cases.

It can be seen from Table 9 that, POA has the highest accuracy performance on all images with four different segmentation levels 2, 4, 6 and 8 in terms of the best values of FSIM. FDRPSO performs well on Lena, Starfish and T1down256 with the segmentation level 4, Butterfly with the segmentation levels 4 and 8 and Skull with the segmentation levels 6 and 8. SaDE performs well on Lena and Baboon with the segmentation level 6 and T1down256 with the segmentation level 8. POA performs well in the other cases.

By analyzing the results in Tables 7-9, the mean values obtained by POA are better than those obtained by other algorithms in many cases. It can be concluded that POA can obtain a good performance in terms of the PSNR, SSIM and FSIM values on the given benchmark images with different segmentation levels, although it is not always the best among all algorithms.

6. Conclusion

A novel population-based optimization method called Poplar Optimization Algorithm (POA) is developed in the paper. By mimicking the sexual and asexual propagation mechanism of poplar, we design execute sexual and asexual propagation for individuals. The performance of POA is evaluated on 25 functions from the CEC2005 test suite and 30 functions from the CEC2017 test suite with different features, where the results are compared with other 14 algorithms. The experimental results indicate that POA algorithm can obtain satisfying optimization performance which is superior to the compared algorithms in most cases. Finally, the optimal segmentation performances also demonstrate the effectiveness of the POA algorithm.

In future works, we will focus on the dynamic principle and convergence analysis of the algorithm. More in-depth study on adaptive topology structures and multi-swarm strategy would make the algorithm more powerful and efficient. Furthermore, the algorithm may be further extended to different research areas such as constrained and dynamic optimization domains. It is also expected that POA would be developed to handle other real-world optimization problems in different application areas.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is partially supported by the National Natural Science Foundation of China (Grant No 61976101). This work is also partially supported by University Natural Science Research Project of Anhui Province (Grant No. KJ2019A0593), the funding plan for scientific research activities of academic and technical leaders and reserve

candidates in Anhui Province (Grant No.2021H264) and the Graduate Innovation Fund of Huaibei Normal University(Grant No. yx2021023).

References

Agrawal, S., Panda, R., Bhuyan, S., & Panigrahi, B. K. (2013). Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm. *Swarm and Evolutionary Computation, 2013*(11), 16–30. https://doi.org/10.1016/j.swevo.2013.02.001

Akay, B., & Karaboga, D. (2012). A modified Artificial Bee Colony algorithm for real–parameter optimization. *Information Sciences, 192*(1), 120–142. https://doi.org/10.1016/j.ins.2010.07.015

Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation, 10*(6), 646–657. https://doi.org/10.1109/TEVC.2006.872133

Chen, D., Zou, F., Lu, R., & Li, S. (2019). Backtracking search optimization algorithm based on knowledge learning. *Information Sciences, 473*, 202–226. https://doi.org/10.1016/j.ins.2018.09.039

Chen, D., Zou, F., Lu, R., & Wang, P. (2017). Learning Backtracking Search Optimisation Algorithm and Its Application. *Information Sciences, 376*, 71–94. https://doi.org/10.1016/j.ins.2016.10.002

Civicioglu, P. (2013). Backtracking search optimization algorithm for numerical optimization problems. *Applied Mathematics and Computation, 219*(15), 8121–8144. https://doi.org/10.1016/j.amc.2013.02.017

Das, S., & Suganthan, P. N. (2011). Differential Evolution: A Survey of the State-of-the-Art. *IEEE Transactions on Evolutionary Computation, 15*(1), 4–31. https://doi.org/10.1109/TEVC.2010.2059031

Dorigo, M. (1992). *Optimization, Learning and Natural Algorithms*. Italy: Politecnico di Milano. Ph.D. Thesis.

He, L., & Huang, S. (2020). An efficient krill herd algorithm for color image multilevel thresholding segmentation problem. *Applied Soft Computing, 2020*(89), Article 106063. https://doi.org/10.1016/j.asoc.2020.106063

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.

Jain, N. K., Nangia, U., & Jain, J. (2018). A Review of Particle Swarm Optimization. *Journal of the Institution of Engineers, 99*(4), 1–5. https://doi.org/10.1007/s40031-018-0323-y

Kaveh, A., & Mahdavi, V. R. (2014). Colliding Bodies Optimization method for optimum discrete design of truss structures. *Computers and Structures, 139*, 43–53. https://doi.org/10.1016/j.compstruc.2014.04.006

Kennedy, J., Eberhart, R. C. (1995). Particle swarm optimization. Proc. of IEEE International Conference on Neural Networks, Piscataway, NJ, 1995, 1942–1948. Doi: 10.1007/978-0-387-30164-8_630.

Kumar, M., Husain, M., Upreti, N., & Gupta, D. (2010). Genetic algorithm: Review and application. *Journal of Information & Knowledge Management, 2*(2), 451–454. https://doi.org/10.2139/ssrn.3529843

Liang, J., Qin, A. K., Suganthan, P. N., Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281–295. Doi: 10.1109/TEVC.2005.857610.

Mendes, R., Kennedy, J., & Neves, J. (2004). The fully informed particle swarm: Simpler, maybe better. *IEEE Transactions on Evolutionary Computation, 8*(3), 204–210. https://doi.org/10.1109/TEVC.2004.826074

Mirjalili, S. (2016). SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowledge-Based Systems, 96*, 120–133. https://doi.org/10.1016/j.knosys.2015.12.022

Peram, T., Veeramachaneni, K., & Mohan, C. K. (2003). Fitness-distance-ratio based particle swarm optimization. In *In Proceedings of the 2003 IEEE Swarm Intelligence Symposium* (pp. 174–181). https://doi.org/10.1109/SIS.2003.1202264

Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009). Differential Evolution Algorithm with Strategy Adaptation for Global Numerical Optimization. *IEEE Transactions on Evolutionary Computation, 13*(2), 398–417. https://doi.org/10.1109/TEVC.2008.927706

Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design, 43*(3), 303–315. https://doi.org/10.1016/j.cad.2010.12.015

Rao, R. V., & Patel, V. (2012). An elitist teaching–learning-based optimization algorithm for solving complex constrained optimization problems. *International Journal of Industrial Engineering Computations, 3*(4), 535–560. https://doi.org/10.5267/j.ijiec.2012.03.007

Storn, R., & Price, K. (1997). Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization, 11*(4), 341–359. https://doi.org/10.1023/A:1008202821328

Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K. , Tiwari, S. (2005). Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Technical Report, Nanyang Technological University, Singapore And KanGAL Report Number 2005005 (Kanpur Genetic Algorithms Laboratory, IIT Kanpur).

Tan, Y., Zhu, Y. (2010). Fireworks Algorithm for Optimization. In: Tan Y., Shi Y., Tan K. C. (eds) Advances in Swarm Intelligence. ICSI 2010. Lecture Notes in Computer Science, vol 6145. Springer, Berlin, Heidelberg. Doi: 10.1007/978-3-642-13495-1_44.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing, 13*, 600–612. https://doi.org/10.1109/TIP.2003.819861

Wu, G., Mallipeddi, R., Suganthan, P. N. (2017). Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization. National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report 2017.

Zhan, Z., Zhang, J., Li, Y., & Chung, H. S. (2009). Adaptive particle swarm optimization. *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics, 39*(6), 1362–1381. https://doi.org/10.1109/TSMCB.2009.2015956

Zhan, Z. H., Zhang, J., Li, Y., & Shi, Y. H. (2011). Orthogonal learning particle swarm optimization. *IEEE Transactions on Evolutionary Computation, 15*(6), 832–847. https://doi.org/10.1109/TEVC.2010.2052054

Zou, F., Chen, D., & Xu, Q. (2019). A Survey of Teaching–Learning-Based Optimization. *Neurocomputing, 335*, 366–383. https://doi.org/10.1016/j.neucom.2018.06.076

Zou, F., Wang, L., Hei, X., Chen, D., & Yang, D. (2014). Teaching–learning-based optimization with dynamic group strategy for global optimization. *Information Sciences, 273*, 112–131. https://doi.org/10.1016/j.ins.2014.03.038