Swarm and Evolutionary Computation 76 (2023) 101212

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Elite archives-driven particle swarm optimization for large scale numerical optimization and its engineering applications

Yiying Zhang

School of Electrical and Information Engineering, Jiangsu University, Zhenjiang, 212013, China

ARTICLE INFO

Keywords:
Particle swarm optimization
Elite archives
Global optimization
Multimodal problems
Metaheuristics

ABSTRACT

Particle swarm optimization (PSO) is a very simple and effective metaheuristic algorithm. Search operators with similar behavior may lead to the loss of diversity in the search space. All particles in PSO have the same and single search strategy. Therefore, PSO may suffer from premature convergence in solving complex multimodal problems. To improve the global search ability of PSO, this paper reports an elite archives-driven particle swarm optimization (EAPSO). Note that, EAPSO only needs population size and terminal condition for performing the search task, which can distinguish EAPSO over the other reported variants of PSO. In addition, EAPSO has a clear structure, which first builds three types of elite archives to save three different hierarchical particles. Then, six learning strategies for updating the positions of particles are designed by reusing these particles of the three elite archives. To verify the performance of EAPSO, EAPSO is employed to solve CEC 2013 test suite with dimensions 30–100 and three constrained engineering problems. Experimental results show that EAPSO outperforms the compared seven powerful variants of PSO on more than half of test functions and offers highly competitive optimal solutions on the considered engineering problems. That is, experimental results support the validity of the improved strategies and prove the superiority of EAPSO in solving complex multimodal problems. The source code of EAPSO can be found by the following website: https://github.com/jsuzyy/EAPSO.

1. Introduction

Particle swarm optimization (PSO) [1] is a very popular population-based algorithm, which is inspired by the flock of birds. PSO has a simple structure and its powerful search ability has been proven. In addition, PSO refers to few control parameters. These characteristics of PSO are very helpful for expanding its applications. Given this, PSO and its variants have been used widely to solve many different types of practical problems since it was developed, such as image classification [2], the production scheduling problem with mold maintenance [3], dynamic job shop scheduling problems [4], plant diseases diagnosis [5], feature selection [6], green coal production problem [7], and cloud workflow scheduling [8].

Note that, all particles in PSO have the same and single search strategy in performing search tasks, which will harm its population diversity. Therefore, PSO may suffer from premature converge (PSO converges to local optimal solutions quickly) in solving complex multimodal problems [9,10]. A lot of variants of PSO have been reported over the past twenty years to improve the global search ability of PSO, whose improved strategies mainly focus on the following four aspects:

- Parameters settings. In the original PSO, there are two control parameters, which are called local acceleration coefficient and global acceleration coefficient. In addition, Shi and Eberhart [11] propose a very popular variant of PSO, which introduce a new parameter called inertia weight to balance global search and local search of PSO. Researchers have reported many variants of PSO by adjusting the three control parameters, such as adaptive particle swam optimization [12] and heterogeneous comprehensive learning particle swam optimizer [13].
- Learning strategy. There is only one learning strategy to update the position of the particle in the original PSO, which is not good for keeping population diversity. To increase the population diversity, some reported variants of PSO employ multiple learning strategies (which offer more search directions for particles) to update the population, such as triple archives particle swam optimization [14], particle swarm optimization with adaptive learning strategy [15], particle swarm optimization based dictionary learning [16], and particle swam optimization based on dimensional learning strategy [17].
- Neighborhood topology. Neighborhood topology is an effective method to keep population diversity by sharing the population information, which has been introduced to many variants of

E-mail address: zhangyiying@ujs.edu.cn.

https://doi.org/10.1016/j.swevo.2022.101212
Received 18 April 2022; Received in revised form 8 November 2022; Accepted 21 November 2022
Available online 23 November 2022
2210-6502/© 2022 Published by Elsevier B.V.

Nomenclature

Abbreviations

AWPSO	Adaptive weighted particle swarm optimizer
CHCLPSO-I	Chaotic heterogeneous comprehensive learning particle swarm optimizer-I
CHCLPSO-II	Chaotic heterogeneous comprehensive learning particle swarm optimizer-II
CHCLPSO-III	Chaotic heterogeneous comprehensive learning particle swarm optimizer-III
EAPSO	Elite archives-driven particle swarm optimization
EPSO	Ensemble particle swarm optimization
GWO	Grey wolf optimizer
HGSPSO	Hybrid gravitational search particle swarm optimizer
PSO	Particle swarm optimization
PSO-w	Particle swarm optimization with inertia weight
QPSO	Quantum particle swarm optimizer

Symbols

c_1	Local acceleration coefficient
c_2	Global acceleration coefficient
D	Dimension
N	Population size
S	The sensitivity of the dimension
t	The current number of iterations
T_{max}	The maximum number of iterations
w	Inertia weight
$\boldsymbol{A}^t$	The first type of elite archive
$\boldsymbol{B}^t$	The second type of elite archive
$\boldsymbol{C}^t$	The third type of elite archive
$\boldsymbol{g}^t$	The optimal position
$\boldsymbol{p}_i^t$	The historical best position of particle i
$\boldsymbol{v}_i^t$	The velocity of particle i
$\boldsymbol{X}^t$	Population
$\boldsymbol{x}_i^t$	The position of particle i
$\boldsymbol{X}_c^t$	common subpopulation
$\boldsymbol{X}_o^t$	Outstanding subpopulation

PSO, such as three-level particle swarm optimization with variable neighborhood search [3], hierarchical dynamic neighborhood based particle swam optimizer [18], and neighborhood-based particle swarm optimizer with discrete crossover [19].

- Hybrid strategy. Hybrid strategy is to enhance the global search ability of PSO by combining the advantages between PSO and other optimization methods, such as hybrid wolf pack algorithm and particle swam optimization [20], hybrid harmony search and particle swarm optimizer [21], and hybrid genetic algorithm and particle swam optimizer [22].

Nature-inspired algorithms perform the search tasks by simulating some natural phenomenon. However, the natural phenomenon usually is very complicated. To make complex natural phenomenon suitable for optimization methods, authors of the reported nature-inspired algorithms usually define some simple rules to simplify the natural phenomenon. Here, it should be pointed out that nearly all the reported nature-inspired algorithms need some control parameters to service for their defined rules, such as scale factor and mix rate in backtracking search algorithm [23], scale factor and crossover rate in differential evolution [24], discovery probability in cuckoo search [25], and inertia weight, local acceleration coefficient and global acceleration coefficient in particle swarm optimization [26]. A common phenomenon is that the control parameters usually are set to different values in solving different optimization problems with different characteristics. Take the variants of PSO as examples, in [27], inertia weight fluctuates between 0.1 and 1.0, local acceleration coefficient is 2.0, global acceleration coefficient is 2.0; in [28], inertia weight is 0.4, global acceleration coefficient is 1.6, local acceleration coefficient is 1.2; in [29], inertia weight can vary between 0.4 and 0.9, local acceleration coefficient is 1.49445, global acceleration coefficient is 1.49445; in [30], inertia weight is 0.7298, global acceleration coefficient is 1.49618 and local acceleration coefficient is 1.49618; in [31], inertia weight is 0.4, local acceleration coefficient is 0.2, global acceleration coefficient is 2. Note that, although these methods with control parameters can get promising results on the considered problems, they also face a great challenge: how to set their control parameters for the unknown optimization problems? In other words, if the control parameters are "right", satisfied solutions will be obtained; otherwise, it will produce bad solutions. In addition, authors usually do not offer the methods of adjusting the control parameters and discuss the impact of the control parameters on the performance of the proposed methods. Besides, the more the number of the control parameters is, the harder the task of adjusting the control parameters is. Thus, the control parameters restrict the applications of these optimization methods.

As previously mentioned, a lot of variants of PSO have been proposed in the past twenty years. However, it is worth mentioning that nearly all the reported variants of PSO need one or more control parameters. Motivated by the mentioned challenges, this paper presents a novel variant of PSO without the control parameters for global optimization problems. The main contributions of this paper can be summarized as follows:

(1) A novel variant of PSO, called elite archives-driven particle swarm optimization (EAPSO), is reported. When solving optimization problems, EAPSO only needs essential parameters, i.e. population size and terminal condition (e.g., the number of function evaluations, the maximum number of iterations, or the given threshold value) and does not refer to the other control parameters, which can distinguish EAPSO over the other reported variants of PSO.
(2) The core of EAPSO is the built three types of elite archives. During each loop, particles are first divided into outstanding particles and common particles based on the fitness value of each particle. The historical best solutions of outstanding particles form the first type of elite archive. The second type of elite archive consists of some promising historical best solutions of all particles. Some promising global best solutions are saved to the third type of elite archive. Outstanding particles can go directly to the next generation population while common particles are optimized by the designed six candidate learning strategies based on the built three types of elite archives.
(3) EAPSO uses inertia weight and compensating factor to balance its local exploitation and global exploration. Inertia weight and compensating factor are very simple in EAPSO, which can be regarded as random vectors consisting of some random numbers between 0 and 1 with uniform distribution.
(4) A series of experiments referred to numerical problems and engineering problems are executed to investigate the performance of EAPSO in solving complex multimodal optimization problems.

The rest of this paper is organized as follows: in Section 2, brief introduction of PSO is given. Section 3 presents the framework and implementation of EAPSO. The experimental results of EAPSO and the compared algorithms on CEC 2013 test suite are shown in Section 4. Section 5 evaluates the performance of EAPSO in solving three real-world engineering design problems. In Section 6, the effectiveness of the improved strategies in EAPSO is discussed. Finally, the conclusion and future research are made in Section 7.

2. Preliminaries

The proposed EAPSO is an improved version of PSO. Thus, this section first introduces two popular versions of PSO, i.e. basic PSO and PSO with inertia weight (PSO-w), which are the basis of EAPSO.

2.1. Basic PSO

PSO is a population-based optimization technique and was developed by Kennedy and Eberhart in 1995 [1]. Each individual in the population is called "particle", which is regarded as a candidate solution. Therefore, the optimization process of PSO can be viewed as the process of continually reaching global optimal solution to the given problem by frequently updating the positions of particles in the population. Updating the position of particle i is associated with four vectors including its historical best position $\boldsymbol{p}_i^t=(p_{i,1}^t,p_{i,2}^t,\dots,p_{i,D}^t)$, its position $\boldsymbol{x}_i^t=(x_{i,1}^t,x_{i,2}^t,\dots,x_{i,D}^t)$, its velocity $\boldsymbol{v}_i^t=(v_{i,1}^t,v_{i,2}^t,\dots,v_{i,D}^t)$, and the global best position $\boldsymbol{g}^t=(g_{i,1}^t,g_{i,2}^t,\dots,g_{i,D}^t)$, where t is the current number of iterations and D is the number of variables. The position of particle i can be updated by

$$x_{i,j}^{t+1}=x_{i,j}^t+v_{i,j}^{t+1}, \tag{1}$$

$$v_{i,j}^{t+1} = v_{i,j}^t + c_1\times r_1\times(p_{i,j}^t - x_{i,j}^t) + c_2\times r_2\times(g_j^t - x_{i,j}^t), \tag{2}$$

where i meeting $i\in[1,N]$ is the index of each particle in the population, N is population size, j meeting $j\in[1,D]$ is the index of each variable, c_1 is local acceleration coefficient, c_2 is global acceleration coefficient, and r_1 and r_2 are two uniformly distributed random numbers between 0 and 1. From Eqs. (1) and (2), the basic idea of PSO can be described as follows. Each particle is attracted toward its own previous best position and the global best position of the population by adjusting its velocity. In addition, the position of particle i is usually initialized by

$$x_{i,j}^t = l_j + r_3\times(u_j - l_j), \tag{3}$$

where r_3 is a uniformly distributed random number between 0 and 1, l_j is the lower boundary of the jth dimension, and u_j is the upper boundary of the jth dimension.

2.2. PSO-w

PSO-w was proposed by Shi and Eberhart in 1998 [11]. The only difference between PSO and PSO-w is that PSO-w introduces a new parameter called inertia weight to update the velocity of the particle. According to PSO-w, Eq. (2) can be rewritten as

$$v_{i,j}^{t+1} = w\times v_{i,j}^t + c_1\times r_1\times(p_{i,j}^t - x_{i,j}^t) + c_2\times r_2\times(g_j^t - x_{i,j}^t), \tag{4}$$

where w is called inertia weight. The introduced w is to balance local search and global search of PSO. From Eq. (4), a larger inertia weight can enhance the global search ability and a smaller inertia weight is helpful for the local search [11]. Given the advantages of inertia weight, at present, most of the reported variants of PSO are based on PSO-w.

3. EAPSO

As mentioned in Section 1, developing the variants of PSO without control parameters is a very valuable topic. To improve the performance of PSO, we focus on overcoming the following two drawbacks of PSO on the basis of avoiding the control parameters:

(1) Lack communication among particles. According to Eqs. (2) and (4), the position of each particle is updated by its velocity, its historical best position and the global best position. Thus, it is lack communication among particles in PSO and PSO-w, which is a potential risk of PSO and PSO-w to fall into the local optimal solutions in solving challenging multimodal optimization problems. Increasing the communication among particles is very helpful for sharing the population information, which can enhance the global search ability of PSO and PSO-w [30,32,33]. To share the population information among particles, three types of elite archives used to save some promising particles are built, which make full use of the overall information of the whole population.

(2) Single learning strategy. As discussed in [34], search operators with similar behavior may lead to the loss of population diversity. Thus, when the obtained best solution is a local optimum, it is a hard task for the population to escape from it. However, by observing Eqs. (2) and (4), either PSO or PSO-w has only one learning strategy, which is a very important factor that PSO and PSO-w may easily get trapped in local optimal solutions for solving complex multimodal problems. To overcome this drawback of PSO, six learning strategies in EAPSO are designed based on the created elite archives to execute the search tasks.

Based on the above discussion, the theoretical foundation of EAPSO is very clear and can be described as follows. The particles guided by the valuable population information have more chances to find the better solutions, which is the theoretical foundation of the original PSO. As shown in Eq. (2), one particle is guided by its historical best position and the current optimal position of the swarm. Motivated by this, EAPSO is aimed at improving the global search ability of PSO by sharing the valuable population information among particles. In EAPSO, the valuable population information is the elite particles saved to three types of elite archives that are the basis of EAPSO in designing six learning strategies of guiding the search directions of particles.

3.1. The framework of EAPSO

The framework of EAPSO has been shown in Fig. 1. In Fig. 1, $\boldsymbol{X}^t$, $\boldsymbol{X}_{\mathrm{o}}^t$, and $\boldsymbol{X}_{\mathrm{c}}^t$ stand for the whole population including all particles at time t, the outstanding subpopulation formed by outstanding particles at time t and the common subpopulation consisting of common particles at time t, respectively. The basic idea of EAPSO is as follows.

$\boldsymbol{X}^t$ $\left(\boldsymbol{X}^t=\left\{\boldsymbol{x}_1^t,\boldsymbol{x}_2^t,\dots,\boldsymbol{x}_N^t\right\}\right)$ is first divided into two subpopulations based on the fitness values of individuals. The half of particles with the better fitness values are called outstanding particles, which form the outstanding subpopulation $\boldsymbol{X}_{\mathrm{o}}^t$ $\left(\boldsymbol{X}_{\mathrm{o}}^t=\left\{\boldsymbol{x}_{\mathrm{o},1}^t,\boldsymbol{x}_{\mathrm{o},2}^t,\dots,\boldsymbol{x}_{\mathrm{o},N/2}^t\right\}\right)$; the rest half of particles with the worse fitness values are called common particles, which make up the common subpopulation $\boldsymbol{X}_{\mathrm{c}}^t$ $\left(\boldsymbol{X}_{\mathrm{c}}^t=\left\{\boldsymbol{x}_{\mathrm{c},1}^t,\boldsymbol{x}_{\mathrm{c},2}^t,\dots,\boldsymbol{x}_{\mathrm{c},N/2}^t\right\}\right)$. When executing the search task, the particles in the $\boldsymbol{X}_{\mathrm{o}}^t$ go directly to the next generation population. The particles in the $\boldsymbol{X}_{\mathrm{c}}^t$ are optimized by the designed six candidate learning strategies. Then, the next generation population $\boldsymbol{X}^{t+1}$ consists of $\boldsymbol{X}_{\mathrm{c}}^{t+1}$ and $\boldsymbol{X}_{\mathrm{o}}^t$. Note that, the designed six candidate learning strategies are based on the built three types of elite archives, i.e. $\boldsymbol{A}^t$, $\boldsymbol{B}^t$, and $\boldsymbol{C}^t$. Obviously, the main components of EAPSO are the built three types of elite archives and the designed six learning strategies, which will be described in detailed in Sections 3.2 and 3.3.

3.2. The built three types of elite archives

This section is to introduce the built three types of elite archives, which are employed to guide the search directions of common particles. In addition, for ease of description, $\boldsymbol{V}_{\mathrm{c}}^t$ $\left(\boldsymbol{V}_{\mathrm{c}}^t=\left\{\boldsymbol{v}_{\mathrm{c},1}^t,\boldsymbol{v}_{\mathrm{c},2}^t,\dots,\boldsymbol{v}_{\mathrm{c},N/2}^t\right\}\right)$ denotes the velocity of $\boldsymbol{X}_{\mathrm{c}}^t$; $\boldsymbol{P}_{\mathrm{o}}^t$ $\left(\boldsymbol{P}_{\mathrm{o}}^t=\left\{\boldsymbol{p}_{\mathrm{o},1}^t,\boldsymbol{p}_{\mathrm{o},2}^t,\dots,\boldsymbol{p}_{\mathrm{o},N/2}^t\right\}\right)$ denotes

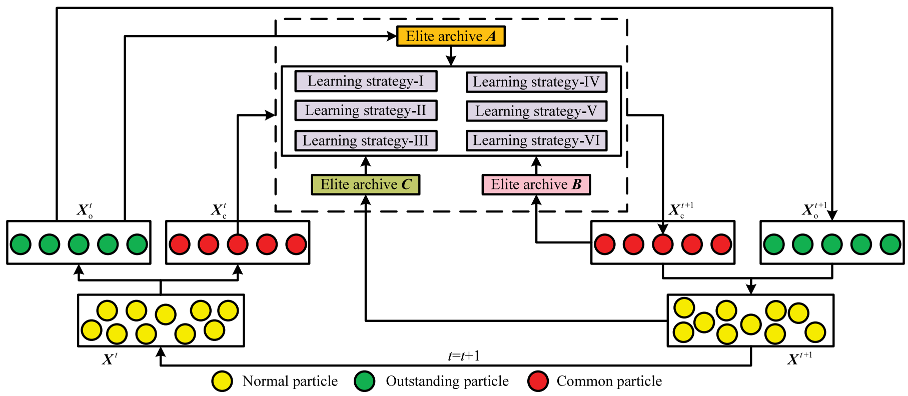


Fig. 1. The framework of the proposed EAPSO.

the historical best solution of $\boldsymbol{X}_{\mathrm{o}}^t$; $\boldsymbol{P}_{\mathrm{c}}^t \left(\boldsymbol{P}_{\mathrm{c}}^t = \left\{\boldsymbol{p}_{\mathrm{c},1}^t, \boldsymbol{p}_{\mathrm{c},2}^t, \ldots, \boldsymbol{p}_{\mathrm{c},N/2}^t\right\}\right)$ denotes the historical best solution of $\boldsymbol{X}_{\mathrm{c}}^t$. Next, the three types of archives, namely $\boldsymbol{A}^t$, $\boldsymbol{B}^t$, and $\boldsymbol{C}^t$, are introduced.

- The first type of elite archive, i.e. $\boldsymbol{A}^t$. From Fig. 1, $\boldsymbol{A}^t$ is from $\boldsymbol{X}_{\mathrm{o}}^t$, which is used to save the historical best solutions of outstanding particles. Thus, $\boldsymbol{A}^t$ can be donated by $\boldsymbol{A}^t = \left\{\boldsymbol{a}_1^t, \boldsymbol{a}_2^t, \ldots, \boldsymbol{a}_{N/2}^t\right\}$ and $\boldsymbol{a}_i^t = \boldsymbol{p}_{\mathrm{o},i}^t$. Obviously, if outstanding subpopulation and common subpopulation have been determined, $\boldsymbol{A}^t$ is also determined and is not changed in one loop.
- The second type of elite archive, i.e. $\boldsymbol{B}^t$. $\boldsymbol{B}^t$ is employed for saving some promising historical best solutions of all particles in the $\boldsymbol{X}^t$, whose maximum length is the same with population size. $\boldsymbol{B}^t$ can be denoted by $\boldsymbol{B}^t = \{\boldsymbol{b}_1^t, \boldsymbol{b}_2^t, \ldots, \boldsymbol{b}_N^t\}$. The individuals in the $\boldsymbol{B}^t$ is dynamic in one loop, which can be explained as follows. As mentioned above, only common particles are optimized in EAPSO. If $\boldsymbol{x}_{\mathrm{c},i}^t$ can find a better solution than its historical best solution $\boldsymbol{p}_{\mathrm{c},i}^t$, $\boldsymbol{B}^t$ will be updated as shown in Algorithm 1.
- The third type of elite archive, i.e. $\boldsymbol{C}^t$. $\boldsymbol{C}^t$ is built for saving some promising global best solutions, whose maximum length is also equal to population size. $\boldsymbol{C}^t$ can be denoted by $\boldsymbol{C}^t = \{\boldsymbol{c}_1^t, \boldsymbol{c}_2^t, \ldots, \boldsymbol{c}_N^t\}$. After completing the tth loop, the global optimal solution at time t is obtained, i.e. $\boldsymbol{g}^t$. Clearly, like $\boldsymbol{B}^t$, the individuals in the $\boldsymbol{C}^t$ is also dynamic in one loop. The method of updating $\boldsymbol{C}^t$ can be found in Algorithm 2.

Algorithm 1 The method of updating $\boldsymbol{B}^t$

```
Input:
   x_{c,i}^{t+1}, p_{c,i}^t, B^t, N, L_B (the length of B^t)
Output:
   B^t, L_B
if f(x_{c,i}^{t+1}) < f(p_{c,i}^t) then
  if L_B < N then
    L_B=L_B+1;
    b_{L_B}^t = x_{c,i}^{t+1};
  else
    Select two individuals, i.e. b_a^t and b_b^t, from B^t randomly;
    if f(x_{c,i}^{t+1}) > f(b_a^t)&&f(x_{c,i}^{t+1}) > f(b_b^t) then
      Break;
    else
      Use x_{c,i}^{t+1} to replace the worse one between b_a^t and b_b^t;
    end if
  end if
else
  Break.
end if
```

Algorithm 2 The method of updating $\boldsymbol{C}^t$

```
Input:
   g^t, C^t, N, L_C (the length of C^t)
Output:
   C^t, L_C
if L_B < N then
  L_C=L_C+1;
  c_{L_C}^t = g^t;
else
  Select two individuals, i.e. c_a^t and c_b^t, from C^t randomly;
  Use g^t to replace the worse one between c_a^t and c_b^t.
end if
```

3.3. The designed learning mechanism

In EAPSO, an individual in each archive is selected to take part in the design of learning strategies. As described in Section 3.2, the length of $\boldsymbol{A}^t$, $\boldsymbol{B}^t$, and $\boldsymbol{C}^t$ is $N/2$, N, and N, respectively. Assume $\boldsymbol{a}_m^t$, $\boldsymbol{b}_r^t$, and $\boldsymbol{c}_q^t$ are the selected particles from $\boldsymbol{A}^t$, $\boldsymbol{B}^t$, and $\boldsymbol{C}^t$, respectively, where $m(m \in [1, N/2])$, $r(r \in [1, N])$, and $q(q \in [1, N])$ are three integers selected randomly. The learning strategies in EAPSO are as follows:

Case 1: $f\left(\boldsymbol{a}_m^t\right) \le f\left(\boldsymbol{b}_r^t\right) \&\& f\left(\boldsymbol{a}_m^t\right) \le f\left(\boldsymbol{c}_q^t\right)$

In this case, the fitness value of $\boldsymbol{a}_m^t$ is superior to those of $\boldsymbol{b}_r^t$ and $\boldsymbol{c}_q^t$. According to this case, common particle i guided by $\boldsymbol{a}_m^t$ and $\boldsymbol{g}^t$, which can be written by

$$v_{\mathrm{c},i,j}^{t+1}=w_{\mathrm{R},j}^t \times v_{\mathrm{c},i,j}^t+\lambda_{1,j}^t \times (a_{m,j}^t - x_{\mathrm{c},i,j}^t) + \lambda_{2,j}^t \times (g_j^t - x_{\mathrm{c},i,j}^t), \tag{5}$$

where $w_{\mathrm{R},j}^t$ is the value of the jth variable of the random weight $\boldsymbol{w}_{\mathrm{R}}^t(\boldsymbol{w}_{\mathrm{R}}^t = [w_{\mathrm{R},1}^t, w_{\mathrm{R},2}^t, \ldots, w_{\mathrm{R},D}^t])$, $\lambda_{1,j}^t$ is the value of the jth variable of the first compensating factor ($\lambda_1^t = [\lambda_{1,1}^t, \lambda_{1,2}^t, \ldots, \lambda_{1,D}^t]$), $\lambda_{2,j}^t$ is the value of the jth variable of the second compensating factor ($\lambda_2^t = [\lambda_{2,1}^t, \lambda_{2,2}^t, \ldots, \lambda_{2,D}^t]$), $a_{m,j}^t$ is the value of the jth variable of $\boldsymbol{a}_m^t$, $v_{\mathrm{c},i,j}^t$ is the value of the jth variable of velocity of the common particle i. In addition, $w_{\mathrm{R},j}^t$, $\lambda_{1,j}^t$, and $\lambda_{2,j}^t$ are three random numbers between 0 and 1 with uniform distribution.

Case 2: $f\left(\boldsymbol{b}_r^t\right) \le f\left(\boldsymbol{a}_m^t\right) \&\& f\left(\boldsymbol{b}_r^t\right) \le f\left(\boldsymbol{c}_q^t\right)$

Clearly, $\boldsymbol{b}_r^t$ has a better fitness value than those of and $\boldsymbol{a}_m^t$ and $\boldsymbol{c}_q^t$ in this case. For this case, common particle i is guided by $\boldsymbol{b}_r^t$ and $\boldsymbol{g}^t$, which can be represented by

$$v_{\mathrm{c},i,j}^{t+1}=w_{\mathrm{R},j}^t \times v_{\mathrm{c},i,j}^t+\lambda_{1,j}^t \times (b_{r,j}^t - x_{\mathrm{c},i,j}^t) + \lambda_{2,j}^t \times (g_j^t - x_{\mathrm{c},i,j}^t), \tag{6}$$

where $b_{r,j}^t$ is the value of the jth variable of $\boldsymbol{b}_r^t$.

Case 3: $f\left(\boldsymbol{c}_q^t\right) \le f\left(\boldsymbol{a}_m^t\right) \&\& f\left(\boldsymbol{c}_q^t\right) \le f\left(\boldsymbol{b}_r^t\right)$

This case demonstrates that $\boldsymbol{c}_q^t$ is the best of $\boldsymbol{a}_m^t$, $\boldsymbol{b}_r^t$, and $\boldsymbol{c}_q^t$ in terms of fitness value. Common particle i is guided by $\boldsymbol{c}_q^t$ and $\boldsymbol{g}^t$ in this case, which can be represented by

$$v_{\mathrm{c},i,j}^{t+1}=w_{\mathrm{R},j}^t \times v_{\mathrm{c},i,j}^t+\lambda_{1,j}^t \times (c_{q,j}^t - x_{\mathrm{c},i,j}^t) + \lambda_{2,j}^t \times (g_j^t - x_{\mathrm{c},i,j}^t), \tag{7}$$

where $c_{q,j}^t$ is the value of the jth variable of $\boldsymbol{c}_q^t$.

Case 4: $f\left(\boldsymbol{c}_q^t\right) \le f\left(\boldsymbol{a}_m^t\right) \&\& f\left(\boldsymbol{b}_r^t\right) \le f\left(\boldsymbol{a}_m^t\right)$

For this case, the fitness values of $\boldsymbol{c}_q^t$ and $\boldsymbol{b}_r^t$ are better than that of $\boldsymbol{a}_m^t$. Thus, $\boldsymbol{c}_q^t$ and $\boldsymbol{b}_r^t$ are selected to guide common particle i to complete the search task, which Can be represented by

$$v_{\mathrm{c},i,j}^{t+1}=w_{\mathrm{R},j}^t \times v_{\mathrm{c},i,j}^t+\lambda_{1,j}^t \times (c_{q,j}^t - x_{\mathrm{c},i,j}^t) + \lambda_{2,j}^t \times (b_{r,j}^t - x_{\mathrm{c},i,j}^t). \tag{8}$$

Case 5: $f\left(\boldsymbol{a}_m^t\right) \le f\left(\boldsymbol{b}_r^t\right) \&\& f\left(\boldsymbol{c}_q^t\right) \le f\left(\boldsymbol{b}_r^t\right)$

Obviously, this case shows that the fitness value of $\boldsymbol{b}_r^t$ cannot compete with those of $\boldsymbol{a}_m^t$ and $\boldsymbol{c}_q^t$. Thus, the search direction of common particle i is guided by $\boldsymbol{a}_m^t$ and $\boldsymbol{c}_q^t$ in this case, which can be represented by

$$v_{\mathrm{c},i,j}^{t+1}=w_{\mathrm{R},j}^t \times v_{\mathrm{c},i,j}^t+\lambda_{1,j}^t \times (a_{m,j}^t - x_{\mathrm{c},i,j}^t) + \lambda_{2,j}^t \times (c_{q,j}^t - x_{\mathrm{c},i,j}^t). \tag{9}$$

Case 6: $f\left(\boldsymbol{a}_m^t\right) \le f\left(\boldsymbol{c}_q^t\right) \&\& f\left(\boldsymbol{b}_r^t\right) \le f\left(\boldsymbol{c}_q^t\right)$

For this case, the fitness value of $\boldsymbol{a}_m^t$ is worse than those of $\boldsymbol{b}_r^t$ and $\boldsymbol{c}_q^t$. Thus, in this case, $\boldsymbol{b}_r^t$ and $\boldsymbol{c}_q^t$ are employed to guide the search process of common particle i, which can be represented by

$$v_{\mathrm{c},i,j}^{t+1}=w_{\mathrm{R},j}^t \times v_{\mathrm{c},i,j}^t+\lambda_{1,j}^t \times (a_{m,j}^t - x_{\mathrm{c},i,j}^t) + \lambda_{2,j}^t \times (b_{r,j}^t - x_{\mathrm{c},i,j}^t). \tag{10}$$

To control the boundary of $v_{\mathrm{c},i,j}^{t+1}$, $v_{\mathrm{c},i,j}^{t+1}$ is adjusted by

$$v_{\mathrm{c},i,j}^{t+1} = \begin{cases} v_{\min,j}, & \text{if } v_{\mathrm{c},i,j}^{t+1} < v_{\min,j}; \\ v_{\max,j}, & \text{if } v_{\mathrm{c},i,j}^{t+1} > v_{\max,j}, \end{cases} \tag{11}$$

where $v_{\min,j}$ and $v_{\max,j}$ are the lower limit and the upper limit of the jth variable of $\boldsymbol{v}_{\mathrm{c},i}^{t+1}$, respectively. Based on the description for six cases shown in Eqs. (5)–(10), the designed learning mechanism for common particle i can be found in Algorithm 3. In Algorithm 3, m_{c}^t is the mean fitness value of all common particles, which can be computed by

$$m_{\mathrm{c}}^t = \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}_{\mathrm{c},i}^t). \tag{12}$$

3.4. The implementation of EAPSO

According to the framework of EAPSO shown in Fig. 1, Algorithm 4 presents the implementation of EAPSO. As can be seen from Algorithm 4, EAPSO has a simple structure and does not require more computational effort. Note that, when solving optimization problems, EAPSO only needs the essential population size and terminal condition and does not refer to other control parameters, which is the most striking feature of EAPSO that can distinguish EAPSO over PSO and the reported variants of PSO.

Computational complexity is a very important indicator to evaluate the computational efficiency of a population-based optimization algorithm. A lower computational complexity means that an algorithm has a higher computational efficiency. From Algorithm 4, the computational complexity of EAPSO is mainly related to the following three aspects:

- Position update. In one loop, the positions of common particles are updated once and the positions of outstanding particles are not changed. So, the complexity of position update can be denoted as $O\left(NDT_{\max}/2\right)$.
- Sort for the population. In one loop, to extract the common particles and the outstanding particles from the population, the population needs to be sorted according to the fitness value. Clearly, the worst complexity of the sort for the population can be written as $O(N^2T_{\max})$.

Algorithm 3 The designed learning mechanism for $\boldsymbol{x}_{\mathrm{c},i}^t$

Input:
$\boldsymbol{x}_{\mathrm{c},i}^t$, $\boldsymbol{v}_{\mathrm{c},i}^t$, $\boldsymbol{a}_m^t$, $\boldsymbol{b}_r^t$, $\boldsymbol{c}_q^t$, $\boldsymbol{w}_{\mathrm{R}}^t$, λ_1^t, and λ_2^t, m_{c}^t
Output:
$\boldsymbol{x}_{\mathrm{c},i}^{t+1}$

if $f(\boldsymbol{x}_{\mathrm{c},i}^t) < m_{\mathrm{c}}^t$ **then**
  if $f\left(\boldsymbol{a}_m^t\right) \le f\left(\boldsymbol{b}_r^t\right) \&\& f\left(\boldsymbol{a}_m^t\right) \le f\left(\boldsymbol{c}_q^t\right)$ **then**
    Compute $\boldsymbol{v}_{\mathrm{c},i}^{t+1}$ by Eq. (5);
  else if $f\left(\boldsymbol{b}_r^t\right) \le f\left(\boldsymbol{a}_m^t\right) \&\& f\left(\boldsymbol{b}_r^t\right) \le f\left(\boldsymbol{c}_q^t\right)$ **then**
    Compute $\boldsymbol{v}_{\mathrm{c},i}^{t+1}$ by Eq. (6);
  else
    Compute $\boldsymbol{v}_{\mathrm{c},i}^{t+1}$ by Eq. (7);
  end if
else
  if $f\left(\boldsymbol{c}_q^t\right) \le f\left(\boldsymbol{a}_m^t\right) \&\& f\left(\boldsymbol{b}_r^t\right) \le f\left(\boldsymbol{a}_m^t\right)$ **then**
    Compute $\boldsymbol{v}_{\mathrm{c},i}^{t+1}$ by Eq. (8);
  else if $f\left(\boldsymbol{a}_m^t\right) \le f\left(\boldsymbol{b}_r^t\right) \&\& f\left(\boldsymbol{c}_q^t\right) \le f\left(\boldsymbol{b}_r^t\right)$ **then**
    Compute $\boldsymbol{v}_{\mathrm{c},i}^{t+1}$ by Eq. (9);
  else
    Compute $\boldsymbol{v}_{\mathrm{c},i}^{t+1}$ by Eq. (10);
  end if
end if
Adjust $\boldsymbol{v}_{\mathrm{c},i}^{t+1}$ by Eq. (11);
Compute $\boldsymbol{x}_{\mathrm{c},i}^{t+1}$ by $\boldsymbol{x}_{\mathrm{c},i}^{t+1} = \boldsymbol{x}_{\mathrm{c},i}^t + \boldsymbol{v}_{\mathrm{c},i}^{t+1}$.

- Comparison of fitness value. In one loop, as can be seen from Algorithm 4, the number of comparison between two individuals is times. Thus, the complexity of the comparison of fitness value can be represented by $O(6.5NT_{\max})$.

Thus, the total computational complexity of EAPSO can be denoted as $O\left(\left(0.5ND + N^2 + 6.5N\right)T_{\max}\right)$.

4. EAPSO on numerical problems

4.1. Experimental preparation

To investigate the optimization performance of EAPSO, EAPSO is employed for solving the well-known CEC 2013 test suite [35] as shown in Table 1. CEC 2013 test suite consists of five unimodal functions (i.e. $\mathrm{F}_1 - \mathrm{F}_5$), 15 simple multimodal functions (i.e. $\mathrm{F}_6 - \mathrm{F}_{20}$) and eight composition functions (i.e. $\mathrm{F}_{21} - \mathrm{F}_{28}$). Obviously, CEC 2013 test suite includes five unimodal functions and 23 multimodal functions. That is, more than 82 percent of test functions are multimodal functions. Multimodal functions usually include multiple local optima and are very difficult to be solved. In addition, the number of feasible solutions grows exponentially as the dimension increases. Therefore, the dimension is also a very important factor of influencing the difficulty of solving an optimization problem. From Table 1, each test function has three candidate dimensions, i.e. 30, 50 and 100. Thus, the applied test suites are very suitable for checking the global search ability of EAPSO. In addition, as the reported Refs. [36–41], the maximum number of function evaluations is set to $5000D$ for all the applied algorithms on each test function. 30 times independently runs are executed on each test function. All the applied algorithms are coded in MATLAB R2017a and the detailed settings of the utilized system have been shown in Table 2.

To check the competitiveness of EAPSO, EAPSO is compared with seven powerful variants of PSO, which include quantum particle swarm optimization (QPSO) [42], ensemble particle swarm optimizer (EPSO) [43], hybrid gravitational search particle swarm optimizer (HGSPSO)

Table 1
The definition for CEC2013 test suite.

No.	Type	Name	Dimension	Range	Optimum
F_1	Unimodal functions	Sphere function	30/50/100	[−100,100]	−1400
F_2		Rotated high conditioned ellipticfunction	30/50/100	[−100,100]	−1300
F_3		Rotated bent cigar function	30/50/100	[−100,100]	−1200
F_4		Rotated discus function	30/50/100	[−100,100]	−1100
F_5		Different powers function	30/50/100	[−100,100]	−1000
F_6	Basic multimodal functions	Rotated rosenbrock's function	30/50/100	[−100,100]	−900
F_7		Rotated schaffers F_7 function	30/50/100	[−100,100]	−800
F_8		Rotated ackley's function	30/50/100	[−100,100]	−700
F_9		Rotated weierstrass function	30/50/100	[−100,100]	−600
F_{10}		Rotated griewank's function	30/50/100	[−100,100]	−500
F_{11}		Rastrigin's function	30/50/100	[−100,100]	−400
F_{12}		Rotated rastrigin's function	30/50/100	[−100,100]	−300
F_{13}		Non-continuous rotated rastrigin's function	30/50/100	[−100,100]	−200
F_{14}		Schwefel's function	30/50/100	[−100,100]	−100
F_{15}		Rotated schwefel's function	30/50/100	[−100,100]	100
F_{16}		Rotated katsuura function	30/50/100	[−100,100]	200
F_{17}		Lunacek bi_rastrigin function	30/50/100	[−100,100]	300
F_{18}		Rotated lunacek bi_rastrigin function	30/50/100	[−100,100]	400
F_{19}		Expanded griewank's plus rosenbrock's function	30/50/100	[−100,100]	500
F_{20}		Expanded scaffer's F_6 function	30/50/100	[−100,100]	600
F_{21}	Composition functions	Composition function 1 (n = 5, rotated)	30/50/100	[−100,100]	700
F_{22}		Composition function 2 (n = 3, unrotated)	30/50/100	[−100,100]	800
F_{23}		Composition function 3 (n = 3, rotated)	30/50/100	[−100,100]	900
F_{24}		Composition function 4 (n = 3, rotated)	30/50/100	[−100,100]	1000
F_{25}		Composition function 5 (n = 3, rotated)	30/50/100	[−100,100]	1100
F_{26}		Composition function 6 (n = 5, rotated)	30/50/100	[−100,100]	1200
F_{27}		Composition function 7 (n = 5, rotated)	30/50/100	[−100,100]	1300
F_{28}		Composition function 8 (n = 5, rotated)	30/50/100	[−100,100]	1400

Algorithm 4 The implementation of EAPSO

Input:
N, D, $t(t{=}0)$, $\boldsymbol{A}^t(\boldsymbol{A}^t{=}\emptyset)$, $\boldsymbol{B}^t(\boldsymbol{B}^t{=}\emptyset)$, $\boldsymbol{C}^t(\boldsymbol{C}^t{=}\emptyset)$, $\boldsymbol{l}$, $\boldsymbol{u}$, $L_{\mathrm{B}}(L_{\mathrm{B}}=0)$, $L_{\mathrm{C}}(L_{\mathrm{C}}=0)$, and $T_{\max}$
Output:
$\boldsymbol{g}^t$

```
Generate X^t by Eq. (3);
Evaluate X^t and get g^t;
p_i^t = x_i^t, v_i^t = 0, i = 1, 2, ..., N;
Select two best particles from X^t into A^t and B^t;
Update L_B and L_C by L_B = L_B + 2 and L_C = L_C + 2;
while t < T_max do
  Compute X_o^t and X_c^t from X^t; compute P_o^t and P_c^t from P^t;
  compute V_c^t from V^t;
  A^t = P_o^t
  for i = 1 to N/2 do
    Select a_m^t, b_r^t, c_q^t from A^t, B^t, and C^t, respectively;
    Generate w_R^t, λ_1^t, and λ_2^t; compute m_c^t by Eq. (12);
    Compute x_{c,i}^{t+1} by Algorithm 3;
    if f(x_{c,i}^{t+1}) < f(p_{c,i}^t) then
      p_{c,i}^t = x_{c,i}^{t+1};
      Update B^t and L_B by Algorithm 1;
      if f(x_{c,i}^{t+1}) < f(g^t) then
        g^t = x_{c,i}^{t+1};
      end if
    end if
  end for
  Update C^t and L_C by Algorithm 2;
  Generate X^{t+1} by aggregating X_o^t and X_c^{t+1}.
end while
```

[44], chaotic heterogeneous comprehensive learning particle swarm optimization-I (CHCLPSO-I) [45], chaotic heterogeneous comprehensive learning particle swarm optimization-II (CHCLPSO-II) [45], chaotic heterogeneous comprehensive learning particle swarm optimization-III (CHCLPSO-III) [45], and adaptive weighted particle swarm optimization (AWPSO) [46]. In addition, Table S1 shows the parameters settings of all algorithms obtained by Taguchi method [47], which can be found in the supplementary material.

Table 2
The detailed settings of the utilized system.

Name	Setting
Hardware	
Solid state drive	500G
RAM	16G
Frequency	2.80 GHz
CPU	11th Gen Intel(R) Core(TM) i7 processor
Software	
Language	MATLAB R2017a
Operating system	Windows 10

4.2. Comparison on solution accuracy

This section is to compare solution accuracy based on the experimental results of EAPSO and the compared algorithms on CEC 2013 test suite. Experimental results are shown in Tables 3, 4, and 5. In the three tables, the best results have been highlighted in bold. In addition, "MEAN" and "STD" denote mean fitness value and standard deviation, respectively. MEAN can be defined by

$$\mathrm{MEAN} = \frac{1}{m}\sum_{k=1}^{m}\left(f(\mathbf{x}_k^*) - f(\mathbf{x}_{\mathrm{R}})\right), \tag{13}$$

where m is the number of runs, $f(\cdot)$ is the objective function, $\boldsymbol{x}_k^*$ is the obtained best solution at kth run by one algorithm, and $\boldsymbol{x}_{\mathrm{R}}$ is the real solution.

From Table 3, for test functions with 30-dimensional, EAPSO can obtain the best MEAN on 13 test functions, which include F_2, F_3, F_6, F_8, F_9, F_{10}, F_{12}, F_{13}, F_{15}, F_{16}, F_{18}, F_{24}, and F_{27}. In addition, EAPSO, CHCLPSO-I, CHCLPSO-II, CHCLPSO-III, and QPSO can get the best MEAN on F_1. EAPSO, CHCLPSO-I, CHCLPSO-II, and CHCLPSO-III can

Table 3
Experimental results of EAPSO and the compared algorithms on CEC 2013 test suite with 30-dimensional.

No.	Indicator	QPSO	EPSO	AWPSO	HGSPSO	CHCLPSO-I	CHCLPSO-II	CHCLPSO-III	EAPSO
F_1	Mean	**2.27E−13**	5.81E+00	4.54E+02	2.28E+03	**2.27E−13**	**2.27E−13**	**2.27E−13**	**2.27E−13**
	STD	8.44E−14	7.93E+00	8.45E+01	2.92E+02	7.31E−14	1.03E−13	**0.00E+00**	2.31E−13
	Symbol	=	+	+	+	=	=	=	/
	Rank	2.80	6.00	7.00	8.00	2.73	2.97	**2.48**	4.02
F_2	Mean	1.97E+07	4.57E+07	2.22E+07	3.87E+07	1.62E+07	1.68E+07	1.70E+07	**1.74E+05**
	STD	9.87E+06	1.61E+07	6.08E+06	9.51E+06	4.17E+06	4.18E+06	5.16E+06	**6.78E+04**
	Symbol	+	+	+	+	+	+	+	/
	Rank	4.17	7.33	4.93	7.10	3.73	3.90	3.83	**1.00**
F_3	Mean	1.01E+08	5.35E+09	2.86E+09	6.37E+09	7.16E+08	7.11E+08	8.01E+08	**2.28E+07**
	STD	1.13E+08	2.15E+09	1.93E+09	2.37E+09	5.23E+08	3.67E+08	5.24E+08	**2.63E+07**
	Symbol	+	+	+	+	+	+	+	/
	Rank	2.00	7.10	6.07	7.57	3.93	3.93	4.23	**1.17**
F_4	Mean	1.98E+04	2.53E+04	**5.91E+03**	8.26E+03	3.08E+04	3.09E+04	3.20E+04	8.58E+03
	STD	4.66E+03	4.78E+03	**1.60E+03**	1.67E+03	5.60E+03	7.50E+03	9.10E+03	4.81E+03
	Symbol	+	+	−	=	+	+	+	/
	Rank	4.47	5.50	**1.37**	2.43	6.60	6.57	6.80	2.27
F_5	Mean	3.98E−09	8.12E+00	1.50E+02	5.29E+02	**1.14E−13**	**1.14E−13**	**1.14E−13**	**1.14E−13**
	STD	3.30E−09	8.48E+00	2.87E+01	8.40E+01	4.33E−13	3.75E−13	4.21E−13	**1.98E−13**
	Symbol	+	+	+	+	+	=	=	/
	Rank	5.00	6.00	7.00	8.00	3.30	2.85	2.55	**1.30**
F_6	Mean	3.88E+01	1.14E+02	1.04E+02	1.95E+02	2.54E+01	2.53E+01	2.88E+01	**7.31E+00**
	STD	1.14E+01	2.55E+01	2.42E+01	2.12E+01	7.90E+00	**6.92E+00**	9.63E+00	1.72E+01
	Symbol	+	+	+	+	=	+	+	/
	Rank	4.53	6.70	6.33	7.97	2.93	2.93	3.37	**1.23**
F_7	Mean	**5.01E+01**	7.77E+01	6.55E+01	7.92E+01	7.74E+01	7.99E+01	8.01E+01	6.99E+01
	STD	1.64E+01	1.91E+01	2.75E+01	1.75E+01	1.27E+01	**1.26E+01**	1.29E+01	2.59E+01
	Symbol	−	=	=	=	=	=	=	/
	Rank	**2.17**	4.93	3.37	5.27	5.10	5.33	5.57	4.27
F_8	Mean	2.10E+01	2.10E+01	2.10E+01	2.10E+01	2.10E+01	2.10E+01	2.10E+01	**2.09E+01**
	STD	5.82E−02	**4.70E−02**	4.78E−02	7.14E−02	5.95E−02	6.08E−02	5.56E−02	6.76E−02
	Symbol	+	+	+	=	+	+	=	/
	Rank	4.57	5.37	4.53	4.70	4.30	4.80	4.50	**3.23**
F_9	Mean	3.70E+01	3.30E+01	2.64E+01	3.19E+01	2.52E+01	2.63E+01	2.69E+01	**1.71E+01**
	STD	3.83E+00	3.04E+00	3.06E+00	1.92E+00	**1.91E+00**	2.06E+00	2.01E+00	3.09E+00
	Symbol	+	+	+	+	+	+	+	/
	Rank	7.63	6.57	3.63	6.43	3.03	3.67	4.00	**1.03**
F_{10}	Mean	1.59E+00	1.31E+02	9.51E+01	3.13E+02	6.74E+00	6.94E+00	5.37E+00	**3.18E−02**
	STD	5.80E−01	3.51E+01	2.03E+01	4.93E+01	2.00E+00	2.08E+00	1.70E+00	**1.83E−02**
	Symbol	+	+	+	+	+	+	+	/
	Rank	2.00	6.80	6.20	8.00	4.17	4.23	3.60	**1.00**
F_{11}	Mean	9.08E+01	2.43E+01	1.91E+02	2.54E+02	**5.68E−14**	**5.68E−14**	**5.68E−14**	3.52E+01
	STD	3.67E+01	6.97E+00	2.21E+01	1.36E+01	4.60E−14	4.09E−14	**2.99E−14**	1.40E+01
	Symbol	+	−	+	+	−	−	−	/
	Rank	5.87	4.30	7.03	7.97	2.25	2.05	**1.70**	4.83
F_{12}	Mean	1.93E+02	2.37E+02	1.99E+02	2.46E+02	9.67E+01	9.80E+01	9.95E+01	**6.94E+01**
	STD	1.57E+01	2.24E+01	1.97E+01	2.28E+01	1.42E+01	**1.25E+01**	1.77E+01	2.71E+01
	Symbol	+	+	+	+	+	+	+	/
	Rank	5.33	7.30	5.83	7.53	2.83	2.67	2.90	**1.60**
F_{13}	Mean	2.00E+02	2.44E+02	2.02E+02	2.54E+02	1.59E+02	1.55E+02	1.65E+02	**1.41E+02**
	STD	**9.04E+00**	1.82E+01	1.26E+01	1.94E+01	1.52E+01	2.08E+01	2.06E+01	3.70E+01
	Symbol	+	+	+	+	+	+	+	/
	Rank	5.33	7.37	5.40	7.57	2.57	2.47	3.13	**2.17**
F_{14}	Mean	6.55E+03	5.18E+02	6.48E+03	7.30E+03	1.10E+01	**9.04E−01**	8.70E+00	1.84E+03
	STD	3.55E+02	2.69E+02	4.73E+02	3.35E+02	2.95E+01	**1.46E+00**	2.99E+01	6.29E+02
	Symbol	+	−	+	+	−	−	−	/
	Rank	6.57	4.00	6.53	7.90	2.73	1.77	**1.50**	5.00
F_{15}	Mean	7.36E+03	5.28E+03	6.83E+03	7.18E+03	4.07E+03	**3.97E+03**	4.01E+03	**3.97E+03**
	STD	**3.41E+02**	6.50E+02	4.03E+02	3.56E+02	4.15E+02	4.00E+02	3.81E+02	6.76E+02
	Symbol	+	+	+	+	=	=	=	/
	Rank	7.60	4.80	6.30	7.07	2.67	**2.50**	2.53	2.53
F_{16}	Mean	2.67E+00	2.14E+00	2.60E+00	2.66E+00	1.50E+00	1.41E+00	1.38E+00	**4.64E−01**
	STD	2.76E−01	4.56E−01	3.32E−01	2.96E−01	2.73E−01	**2.44E−01**	2.89E−01	3.89E−01
	Symbol	+	+	+	+	+	+	+	/

(*continued on next page*)

Table 3 (*continued*).

No.	Indicator	QPSO	EPSO	AWPSO	HGSPSO	CHCLPSO-I	CHCLPSO-II	CHCLPSO-III	EAPSO
	Rank	6.77	5.20	6.73	6.93	3.23	3.20	2.80	**1.13**
F_{17}	Mean	2.02E+02	2.01E+02	2.65E+02	3.77E+02	3.13E+01	3.06E+01	**3.05E+01**	7.55E+01
	STD	1.99E+01	3.49E+01	1.53E+01	1.68E+01	4.93E−01	**9.88E−02**	1.02E−01	1.14E+01
	Symbol	+	+	+	+	−	−	−	/
	Rank	5.60	5.47	6.93	8.00	2.97	1.97	**1.07**	4.00
F_{18}	Mean	2.29E+02	3.29E+02	2.63E+02	3.59E+02	2.01E+02	1.94E+02	1.78E+02	**9.19E+01**
	STD	1.73E+01	2.23E+01	1.65E+01	2.59E+01	**1.38E+01**	1.93E+01	1.43E+01	1.69E+01
	Symbol	+	+	+	+	+	+	+	/
	Rank	4.90	7.03	5.93	7.93	3.53	3.13	2.53	**1.00**
F_{19}	Mean	1.52E+01	2.42E+01	2.24E+01	4.42E+01	1.93E+00	1.70E+00	**1.42E+00**	3.52E+00
	STD	2.16E+00	5.29E+00	1.70E+00	6.45E+00	2.86E−01	**2.03E−01**	2.20E−01	1.31E+00
	Symbol	+	+	+	+	−	−	−	/
	Rank	5.07	6.57	6.37	8.00	2.70	2.27	**1.20**	3.83
F_{20}	Mean	**1.26E+01**	1.28E+01	1.31E+01	1.42E+01	1.39E+01	1.40E+01	1.42E+01	1.45E+01
	STD	**3.48E−01**	3.86E−01	1.06E+00	1.14E+00	8.99E−01	7.91E−01	6.67E−01	1.04E+00
	Symbol	−	−	−	=	−	−	−	/
	Rank	**2.30**	3.07	3.50	5.58	4.82	5.02	5.30	6.42
F_{21}	Mean	2.96E+02	4.75E+02	5.87E+02	9.72E+02	2.27E+02	**2.19E+02**	2.32E+02	2.91E+02
	STD	7.23E+01	7.20E+01	7.49E+01	1.48E+02	3.81E+01	**3.69E+01**	4.20E+01	8.30E+01
	Symbol	=	+	+	+	−	−	−	/
	Rank	3.68	6.13	6.83	8.00	2.83	**2.60**	2.97	2.95
F_{22}	Mean	6.66E+03	5.97E+02	6.38E+03	7.52E+03	1.08E+02	1.12E+02	**1.02E+02**	2.45E+03
	STD	5.24E+02	2.90E+02	6.65E+02	4. 0E+02	**2.53E+01**	3.87E+01	2.88E+01	5.98E+02
	Symbol	+	−	+	+	−	−	−	/
	Rank	6.73	3.97	6.37	7.90	2.50	2.00	**1.57**	4.97
F_{23}	Mean	7.60E+03	6.58E+03	7.31E+03	7.71E+03	4.87E+03	4.90E+03	**4.71E+03**	4.80E+03
	STD	**3.31E+02**	6.92E+02	4.15E+02	6.48E+02	4.80E+02	4.68E+02	5.93E+02	9.04E+02
	Symbol	+	+	+	+	=	=	=	/
	Rank	7.03	5.07	6.50	7.23	2.67	2.60	**2.43**	2.47
F_{24}	Mean	2.56E+02	2.66E+02	2.75E+02	2.84E+02	2.68E+02	2.70E+02	2.71E+02	**2.55E+02**
	STD	1.53E+01	1.01E+01	1.16E+01	1.06E+01	7.90E+00	**5.58E+00**	7.46E+00	1.02E+01
	Symbol	=	+	+	+	+	+	+	/
	Rank	2.53	3.97	5.80	7.20	4.27	4.67	5.30	**2.27**
F_{25}	Mean	**2.64E+02**	3.01E+02	2.97E+02	3.09E+02	2.95E+02	2.95E+02	2.96E+02	2.75E+02
	STD	1.17E+01	1.16E+01	9.84E+00	9.56E+00	**4.84E+00**	6.43E+00	5.12E+00	1.11E+01
	Symbol	−	+	+	+	+	+	+	/
	Rank	**1.47**	5.93	5.17	7.03	4.53	4.87	5.10	1.90
F_{26}	Mean	2.54E+02	2.03E+02	3.16E+02	3.26E+02	**2.01E+02**	**2.01E+02**	**2.01E+02**	3.09E+02
	STD	8.30E+01	7.76E−01	7.66E+01	8.13E+01	**4.64E−01**	6.64E−01	5.01E−01	6.72E+01
	Symbol	=	−	+	+	−	−	−	/
	Rank	3.50	5.07	6.23	7.60	2.97	3.13	**2.90**	4.60
F_{27}	Mean	9.90E+02	9.82E+02	9.80E+02	1.13E+03	9.35E+02	9.09E+02	9.89E+02	**8.42E+02**
	STD	1.47E+02	1.49E+02	1.03E+02	**6.44E+01**	1.89E+02	2.35E+02	1.18E+02	8.77E+01
	Symbol	+	+	+	+	+	=	+	/
	Rank	4.80	4.40	4.37	7.30	4.20	4.43	4.50	**2.00**
F_{28}	Mean	**3.00E+02**	7.69E+02	9.17E+02	1.53E+03	**3.00E+02**	**3.00E+02**	**3.00E+02**	4.11E+02
	STD	**1.08E−06**	1.31E+02	2.30E+02	6.31E+01	2.67E−05	6.53E−04	4.09E−04	3.38E+02
	Symbol	+	+	+	+	+	+	+	/
	Rank	2.57	6.10	6.77	7.90	3.53	3.50	4.00	**1.63**
+/ = /−		21/4/3	22/1/5	25/1/2	24/4/0	15/5/8	14/6/8	14/6/8	/
Avg. (Rank)		4.54	5.64	5.68	7.15	3.49	3.43	3.37	**2.71**

achieve the best MEAN on F_5. Thus, EAPSO can get or share the best MEAN on 15 test functions including F_1, F_2, F_3, F_5, F_6, F_8, F_9, F_{10}, F_{12}, F_{13}, F_{15}, F_{16}, F_{18}, F_{24}, and F_{27}. Besides, in terms of MEAN, QPSO, AWPSO, CHCLPSO-II, and CHCLPSO-III are best on two test functions (i.e. F_7 and F_{25}), one test function (i.e. F_4), four test functions (i.e. F_{14}, F_{15}, F_{17}, and F_{21}), and three test functions (i.e. F_{19}, F_{22}, and F_{23}), respectively. CHCLPSO-I, CHCLPSO-II, and CHCLPSO-III can beat the other algorithms on F_{11} and F_{26}. QPSO can find the best MEAN on F_{20}. For F_{28}, QPSO, CHCLPSO-I, CHCLPSO-II, and CHCLPSO-III can find the better MEAN than the other four algorithms.

As presented in Table 4, for test functions with 50-dimensional, EAPSO is superior to the compared algorithms on half of test functions, i.e. F_2, F_3, F_4, F_6, F_9, F_{10}, F_{12}, F_{15}, F_{16}, F_{18}, F_{23}, F_{24}, F_{25}, and F_{27}. In addition, CHCLPSO-I, CHCLPSO-II, CHCLPSO-III, and EAPSO can get better MEAN than the rest four algorithms on F_1. CHCLPSO-I, CHCLPSO-III, and EAPSO can achieve better MEAN than the rest five algorithms on F_5. For F_8, the MEAN of QPSO, HGSPSO, CHCLPSO-I, CHCLPSO-III, and EAPSO is the best. In other words, EAPSO can achieve or share the best MEAN on 17 test functions, i.e. F_1, F_2, F_3, F_4, F_5, F_6, F_8, F_9, F_{10}, F_{12}, F_{15}, F_{16}, F_{18}, F_{23}, F_{24}, F_{25}, and F_{27}. Besides, AWPSO, CHCLPSO-II, and CHCLPSO-III outperform the rest algorithms on two test function (i.e. F_7 and F_{20}), two test functions (i.e. F_{14} and F_{22}), and one test function (i.e. F_{19}), respectively. CHCLPSO-I, CHCLPSO-II, CHCLPSO-III can get the best MEAN on F_{11}, F_{26}, and F_{28}. CHCLPSO-II and CHCLPSO-III have the best MEAN on F_{17}. For F_{21}, the best MEAN is found by CHCLPSO-I and CHCLPSO-III.

By observing the results from Table 5, for test functions with 100-dimensional, EAPSO can offer the better MEAN than the compared

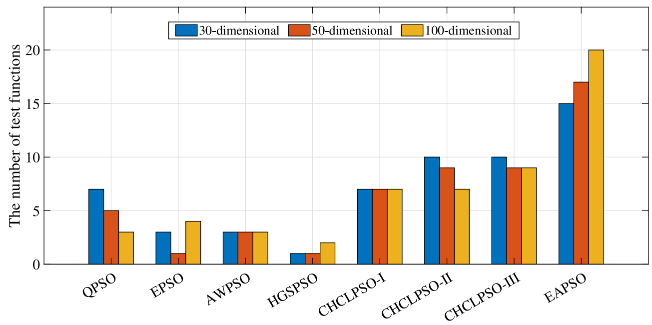


Fig. 2. The number of test functions with the best MEAN obtained by the applied algorithms on CEC 2013 test suite.

algorithms on more than half of test functions, which are F_2, F_3, F_6, F_7, F_9, F_{10}, F_{12}, F_{13}, F_{15}, F_{16}, F_{18}, F_{21}, F_{23}, F_{24}, F_{25}, and F_{27}. In addition, all algorithms can find the same MEAN on F_8 and F_{20}. Thus, EAPSO can get or share the best MEAN on 18 test functions., i.e. F_2, F_3, F_6, F_7, F_8, F_9, F_{10}, F_{12}, F_{13}, F_{15}, F_{16}, F_{18}, F_{20}, F_{21}, F_{23}, F_{24}, F_{25}, and F_{27}. Besides, for F_1, CHCLPSO-I and CHCLPSO-III can get the best MEAN. CHCLPSO-II and CHCLPSO-III can beat the rest algorithms on F_{17} and F_{26}. AWPSO, CHCLPSO-I, and CHCLPSO-III can achieve the best MEAN on one test functions (i.e. F_4), three test functions (i.e. F_5, F_{11}, F_{14}, and F_{28}), and two test functions (i.e. F_{19} and F_{22}), respectively.

To further compare the performance differences among the applied algorithms, Fig. 2 shows the number of test functions with the best MEAN obtained by the applied algorithms on CEC 2013 test suite. From Fig. 2, EAPSO has obvious advantages over the compared algorithms on CEC 2013 test suite with each considered dimension. Although CHCLPSO-I, CHCLPSO-II, and CHCLPSO-III show strong competitiveness, they still cannot compete with EAPSO. In addition, for EAPSO, the number of test functions with the best MEAN increases with the increasing of the dimension, which demonstrates that EAPSO is superior to the compared algorithms in solving high-dimensional problems.

4.3. Comparison on significance test

This section is to determine whether there are significance differences between the results of EAPSO and the compared algorithms by conducting Wilcoxon signed-rank test. Thus, the mean results obtained from 30 runs independently for every algorithm are subjected to Wilcoxon signed-rank test with the level of significance α=0.05. The test results are presented in Tables 3, 4, and 5. In the three tables, symbol "+" means that EAPSO has a better performance than the compared algorithm; symbol "=" indicates that EAPSO has the same performance with the compared algorithm; symbol "−" denotes that EAPSO has a worse performance than the compared algorithm.

As can be seen from Table 3, EAPSO has obvious advantages over QPSO, EPSO, AWPSO, and HGSPSO. EAPSO is superior to QPSO, EPSO, AWPSO, and HGSPSO on 21 test functions, 22 test functions, 25 test functions, and 24 test functions, respectively. However, QPSO, EPSO, and AWPSO only outperform EAPSO on three test functions (i.e. F_7, F_{20}, and F_{25}), five test functions (i.e. F_{11}, F_{14}, F_{20}, F_{22}, and F_{26}), and two test functions (i.e. F_4 and F_{20}), respectively. In addition, CHCLPSO-I, CHCLPSO-II, and CHCLPSO-III can beat EAPSO on eight test functions, i.e. F_{11}, F_{14}, F_{17}, F_{19}, F_{20}, F_{21}, F_{22}, and F_{26}. However, CHCLPSO-I, CHCLPSO-II, and CHCLPSO-III are inferior to EASPO on 15 test functions, 14 test functions, and 14 test functions, respectively. Besides, QPSO, EPSO, AWPSO, HGSPSO, CHCLPSO-I, CHCLPSO-II, and CHCLPSO-III can get the same performance with EAPSO on four test functions (i.e. F_1, F_{21}, F_{24}, and F_{26}), one test function (i.e. F_7), one test function (i.e. F_7), four test functions (i.e. F_4, F_7, F_8, and F_{20}), five test functions (i.e. F_1, F_6, F_7, F_{15}, and F_{23}), six test functions (i.e. F_1, F_5, F_7, F_{15}, F_{23}, and F_{27}), and six test functions (i.e. F_1, F_5, F_7, F_8, F_{15}, and F_{23}), respectively.

As can be seen from Table 4, EAPSO is superior to QPSO, EPSO, AWPSO, HGSPSO, CHCLPSO-I, CHCLPSO-II, and CHCLPSO-III on more than sixty percent of test functions. Further, QPSO, AWPSO, and HGSPSO only beat EAPSO on one test function (i.e. F_{20}), two test functions (i.e. F_4 and F_{20}), and one test function (i.e. F_{20}), respectively. EPSO, CHCLPSO-I, CHCLPSO-II, and CHCLPSO-III have strong competitiveness, which outperforms EAPSO on five test functions (i.e. F_{11}, F_{14}, F_{20}, F_{22}, and F_{26}), eight test functions (i.e. F_{11}, F_{14}, F_{17}, F_{19}, F_{20}, F_{21}, F_{22}, and F_{26}), eight test functions (i.e. F_{11}, F_{14}, F_{17}, F_{19}, F_{20}, F_{21}, F_{22}, and F_{26}), and eight test functions (i.e. F_{11}, F_{14}, F_{17}, F_{19}, F_{20}, F_{21}, F_{22}, and F_{26}), respectively. In addition, QPSO, EPSO, AWPSO, HGSPSO, CHCLPSO-I, CHCLPSO-II, and CHCLPSO-III can get the same performance with EAPSO on five test functions (i.e. F_1, F_8, F_{21}, F_{24}, and F_{25}), one test function (i.e. F_{28}), one test function (i.e. F_7), two test functions (i.e. F_4 and F_8), three test functions (i.e. F_1, F_5, and F_8), one test function (i.e. F_1), and three test functions (i.e. F_1, F_8, and F_{28}), respectively.

Looking at Table 5, EAPSO can find the better solutions than QPSO, EPSO, AWPSO, HGSPSO, CHCLPSO-I, CHCLPSO-II, and CHCLPSO-III on more than sixty percent of test functions. Further, QPSO and HGSPSO cannot beat EAPSO on any functions, which only achieve the same performance with EAPSO on F_{20}. AWPSO only can win EAPSO on F_4 while it is inferior to EAPSO on 24 test functions. EAPSO outperforms EPSO on 22 test functions while EPSO only can offer better solutions than EAPSO on four test functions, i.e. F_{11}, F_{14}, F_{17}, and F_{22}. Note that, unlike QPSO, EPSO, HGSPSO, and AWPSO, CHCLPSO-I, CHCLPSO-II, and CHCLPSO-III show excellent global search ability, which can obtain better solutions than EAPSO on seven test functions (i.e. F_5, F_{11}, F_{14}, F_{17}, F_{19}, F_{22}, and F_{26}), seven test functions (i.e. F_1, F_{11}, F_{14}, F_{17}, F_{19}, F_{22}, and F_{26}), and seven test functions (i.e. F_1, F_{11}, F_{14}, F_{17}, F_{19}, F_{22}, and F_{26}), respectively.

Clearly, EAPSO shows better performance than the compared algorithms on at least half of test functions with each dimension. This supports the excellent global search ability of EAPSO.

4.4. Comparison on ranking results

This section is to compare the performance of EAPSO and the compared algorithms according to ranking. Tables 3, 4, and 5 display the ranking results produced by Friedman test on the mean results of 30 runs dependently. In the three tables, a smaller ranking result indicates a higher ranking. The best results in the three tables are in bold.

From Table 3, EAPSO is the best of all algorithms on half of test functions, i.e. F_2, F_3, F_5, F_6, F_8, F_9, F_{10}, F_{12}, F_{13}, F_{16}, F_{18}, F_{24}, F_{27}, and F_{28}. In addition, QPSO and CHCLPSO-III can take the first place on three test functions (i.e. F_7, F_{20}, and F_{25}) and eight test functions (i.e. F_1, F_{11}, F_{14}, F_{17}, F_{19}, F_{22}, F_{23}, and F_{26}), respectively. For F_4, AWPSO can get the best solution. For F_{15} and F_{21}, CHCLPSO-II offers the better solutions than the rest algorithms. Besides, HGSPSO, EPSO, and CHCLPSO-I cannot achieve the best solutions on any test functions. According to the last column of Table 3, all the applied algorithms can be sorted in the following order (from best to worst): EAPSO, CHCLPSO-III, CHCLPSO-II, CHCLPSO-I, QPSO, EPSO, AWPSO, and HGSPSO. That is, EAPSO is the best of all algorithms.

By observing Table 4, EAPSO outperforms the compared algorithms on more than half of test functions, which include F_2, F_3, F_5, F_6, F_8, F_9, F_{10}, F_{12}, F_{13}, F_{15}, F_{16}, F_{18}, F_{23}, F_{24}, F_{25}, F_{27}, and F_{28}. In addition, CHCLPSO-III can obtain the best solutions on seven test functions, i.e. F_1, F_{11}, F_{14}, F_{17}, F_{19}, F_{21}, and F_{22}. CHCLPSO-I and CHCLPSO-II can get the best solutions on F_{26} and F_{28}, respectively. For F_4, F_7, and F_{20}, AWPSO has advantages over the other algorithms. Besides, QPSO, EPSO, and HGSPSO cannot get the first place on any test functions. As shown in the last column of Table 4, all algorithms can be sorted in the

Table 4
Experimental results of EAPSO and the compared algorithms on CEC 2013 test suite with 30-dimensional.

No.	Indicator	QPSO	EPSO	AWPSO	HGSPSO	CHCLPSO-I	CHCLPSO-II	CHCLPSO-III	EAPSO
F_1	Mean	2.41E−09	5.79E+00	1.48E+03	6.53E+03	**2.27E−13**	**2.27E−13**	**2.27E−13**	**2.27E−13**
	STD	6.00E−09	9.70E+00	1.96E+02	5.76E+02	2.53E−13	2.31E−13	**1.79E−13**	4.96E−13
	Symbol	+	+	+	+	=	=	=	/
	Rank	5.00	6.00	7.00	8.00	2.32	2.32	**1.67**	3.70
F_2	Mean	3.66E+07	1.05E+08	6.55E+07	1.33E+08	3.37E+07	3.51E+07	3.45E+07	**5.25E+05**
	STD	1.31E+07	2.89E+07	1.31E+07	3.07E+07	5.57E+06	8.66E+06	6.25E+06	**1.65E+05**
	Symbol	+	+	+	+	+	+	+	/
	Rank	3.60	7.20	6.00	7.73	3.17	3.67	3.63	**1.00**
F_3	Mean	4.91E+09	2.58E+10	9.22E+09	2.94E+10	5.40E+09	6.62E+09	5.65E+09	**8.84E+07**
	STD	3.25E+09	6.38E+09	4.39E+09	9.06E+09	2.45E+09	2.21E+09	2.44E+09	**8.08E+07**
	Symbol	+	+	+	+	+	+	+	/
	Rank	3.07	7.37	5.10	7.60	3.70	4.43	3.73	**1.00**
F_4	Mean	4.03E+04	4.59E+04	8.66E+03	2.00E+04	3.90E+04	3.88E+04	3.85E+04	**1.93E+04**
	STD	5.62E+03	6.76E+03	**1.34E+03**	2.38E+03	7.23E+03	7.97E+03	8.30E+03	1.25E+04
	Symbol	+	+	−	=	+	+	+	/
	Rank	5.83	6.97	**1.20**	2.60	5.53	5.53	5.57	2.77
F_5	Mean	8.12E−05	2.17E+00	3.46E+02	1.65E+03	**2.27E−13**	3.41E−13	**2.27E−13**	**2.27E−13**
	STD	5.91E−05	2.84E+00	3.42E+01	2.79E+02	5.10E−13	4.92E−13	5.99E−13	**3.76E−13**
	Symbol	+	+	+	+	=	+	+	/
	Rank	5.00	6.00	7.00	8.00	2.57	3.37	2.62	**1.45**
F_6	Mean	4.79E+01	1.79E+02	1.49E+02	3.69E+02	4.60E+01	4.60E+01	4.65E+01	**4.33E+01**
	STD	**9.89E−01**	5.86E+01	4.55E+01	5.27E+01	1.08E+00	1.05E+00	1.02E+00	1.08E+01
	Symbol	+	+	+	+	+	+	+	/
	Rank	4.43	6.77	6.23	8.00	2.97	2.90	3.43	**1.27**
F_7	Mean	9.64E+01	1.17E+02	**8.42E+01**	1.30E+02	1.17E+02	1.16E+02	1.23E+02	8.65E+01
	STD	2.61E+01	1.30E+01	2.27E+01	2.29E+01	1.12E+01	1.13E+01	**1.04E+01**	2.08E+01
	Symbol	=	+	=	+	+	+	+	/
	Rank	3.60	5.10	**2.30**	6.33	5.20	5.10	6.00	2.37
F_8	Mean	**2.11E+01**	2.12E+01	2.12E+01	**2.11E+01**	**2.11E+01**	2.12E+01	**2.11E+01**	**2.11E+01**
	STD	5.44E−02	3.78E−02	3.94E−02	4.27E−02	4.19E−02	**3.53E−02**	3.98E−02	5.16E−02
	Symbol	=	+	+	=	=	+	=	/
	Rank	4.43	4.50	4.87	4.40	4.20	5.17	4.67	**3.77**
F_9	Mean	7.25E+01	6.26E+01	5.02E+01	6.07E+01	5.20E+01	5.32E+01	5.40E+01	**3.72E+01**
	STD	3.10E+00	3.61E+00	4.17E+00	2.55E+00	2.51E+00	**2.29E+00**	2.96E+00	5.37E+00
	Symbol	+	+	+	+	+	+	+	/
	Rank	7.97	6.63	2.83	6.33	3.27	3.77	4.17	**1.03**
F_{10}	Mean	1.73E+01	6.20E+02	3.19E+02	1.08E+03	2.63E+01	2.52E+01	1.96E+01	**5.47E−02**
	STD	9.03E+00	1.54E+02	5.22E+01	1.61E+02	6.79E+00	7.05E+00	5.19E+00	**3.28E−02**
	Symbol	+	+	+	+	+	+	+	/
	Rank	2.83	7.03	6.00	7.97	4.17	4.00	3.00	**1.00**
F_{11}	Mean	2.07E+02	3.35E+01	3.98E+02	5.15E+02	**5.68E−14**	**5.68E−14**	**5.68E−14**	9.35E+01
	STD	6.24E+01	1.03E+01	2.90E+01	2.97E+01	8.38E−14	8.96E−14	**7.24E−14**	2.60E+01
	Symbol	+	−	+	+	−	−	−	/
	Rank	5.97	4.00	7.00	8.00	2.05	2.20	**1.75**	5.03
F_{12}	Mean	4.06E+02	5.00E+02	4.25E+02	5.27E+02	2.61E+02	2.61E+02	2.92E+02	**1.55E+02**
	STD	**2.20E+01**	2.41E+01	2.86E+01	4.20E+01	3.69E+01	3.88E+01	3.41E+01	2.97E+01
	Symbol	+	+	+	+	+	+	+	/
	Rank	5.30	7.23	5.73	7.73	2.87	2.67	3.47	**1.00**
F_{13}	Mean	4.08E+02	5.05E+02	4.26E+02	5.16E+02	4.01E+02	4.00E+02	4.08E+02	**3.13E+02**
	STD	**1.69E+01**	3.41E+01	2.24E+01	3.75E+01	3.74E+01	3.41E+01	3.81E+01	5.65E+01
	Symbol	+	+	+	+	+	+	+	/
	Rank	3.83	7.13	4.77	7.67	3.80	3.40	4.00	**1.40**
F_{14}	Mean	1.25E+04	4.98E+02	1.22E+04	1.39E+04	1.32E+01	**2.04E+00**	5.52E+00	3.37E+03
	STD	4.95E+02	2.38E+02	7.13E+02	3.70E+02	2.28E+01	**1.23E+00**	2.16E+01	9.39E+02
	Symbol	+	−	+	+	−	−	−	/
	Rank	6.70	4.00	6.33	7.97	2.87	1.73	**1.40**	5.00
F_{15}	Mean	1.41E+04	9.51E+03	1.39E+04	1.42E+04	8.35E+03	8.38E+03	8.35E+03	**7.38E+03**
	STD	**4.74E+02**	7.38E+02	5.22E+02	5.98E+02	5.70E+02	5.15E+02	6.89E+02	1.02E+03
	Symbol	+	+	+	+	+	+	+	/
	Rank	7.20	4.60	6.60	7.20	2.83	2.83	3.10	**1.63**
F_{16}	Mean	3.50E+00	2.46E+00	3.54E+00	3.51E+00	1.96E+00	1.79E+00	1.77E+00	**5.73E−01**
	STD	3.84E−01	4.49E−01	**2.88E−01**	3.36E−01	3.50E−01	3.08E−01	3.21E−01	3.44E−01
	Symbol	+	+	+	+	+	+	+	/
	Rank	6.90	4.67	7.00	7.00	3.50	2.93	2.93	**1.07**

(*continued on next page*)

Table 4 (*continued*).

No.	Indicator	QPSO	EPSO	AWPSO	HGSPSO	CHCLPSO-I	CHCLPSO-II	CHCLPSO-III	EAPSO
F_{17}	Mean	4.09E+02	3.21E+02	5.35E+02	7.90E+02	5.22E+01	5.10E+01	**5.09E+01**	1.52E+02
	STD	3.13E+01	4.26E+01	2.03E+01	4.07E+01	6.96E−01	**7.58E−02**	8.08E−02	2.70E+01
	Symbol	+	+	+	+	−	−	−	/
	Rank	5.97	5.03	7.00	8.00	3.00	1.73	**1.27**	4.00
F_{18}	Mean	4.60E+02	6.60E+02	5.38E+02	7.68E+02	4.20E+02	4.05E+02	3.99E+02	**1.74E+02**
	STD	2.15E+01	3.08E+01	**1.60E+01**	4.28E+01	3.84E+01	3.78E+01	2.88E+01	3.07E+01
	Symbol	+	+	+	+	+	+	+	/
	Rank	4.63	7.03	6.00	7.97	3.50	3.13	2.73	**1.00**
F_{19}	Mean	3.03E+01	3.08E+01	4.87E+01	2.53E+02	3.18E+00	2.73E+00	**2.00E+00**	7.99E+00
	STD	3.15E+00	1.37E+01	3.31E+00	9.86E+01	4.64E−01	**4.15E−01**	4.76E−01	2.14E+00
	Symbol	+	+	+	+	−	−	−	/
	Rank	5.70	5.40	6.90	8.00	2.80	2.07	**1.13**	4.00
F_{20}	Mean	2.25E+01	2.27E+01	**2.21E+01**	2.35E+01	2.37E+01	2.37E+01	2.38E+01	2.44E+01
	STD	**3.42E−01**	3.48E−01	6.15E−01	1.07E+00	6.30E−01	7.02E−01	8.43E−01	1.33E+00
	Symbol	−	−	−	−	−	−	−	/
	Rank	2.43	3.20	**1.87**	5.23	5.55	5.57	5.33	6.82
F_{21}	Mean	7.35E+02	9.35E+02	1.26E+03	1.84E+03	**2.52E+02**	2.72E+02	**2.52E+02**	7.46E+02
	STD	4.52E+02	3.56E+02	3.07E+02	4.13E+02	**1.14E+02**	1.52E+02	1.27E+02	3.37E+02
	Symbol	=	+	+	+	−	−	−	/
	Rank	4.13	5.50	6.63	7.80	2.83	2.87	**2.c33**	3.90
F_{22}	Mean	1.26E+04	5.03E+02	1.30E+04	1.45E+04	3.92E+01	**2.52E+01**	3.25E+01	5.22E+03
	STD	6.65E+02	2.64E+02	1.02E+03	7.62E+02	4.75E+01	**1.42E+01**	3.76E+01	1.23E+03
	Symbol	+	−	+	+	−	−	−	/
	Rank	6.37	3.97	6.93	7.70	2.47	1.87	**1.70**	5.00
F_{23}	Mean	1.43E+04	1.19E+04	1.44E+04	1.46E+04	9.91E+03	1.01E+04	1.01E+04	**9.04E+03**
	STD	**3.64E+02**	1.00E+03	6.23E+02	4.90E+02	8.26E+02	5.48E+02	9.03E+02	1.42E+03
	Symbol	+	+	+	+	+	+	+	/
	Rank	6.67	4.77	6.87	7.47	2.77	2.80	3.03	**1.63**
F_{24}	Mean	3.15E+02	3.46E+02	3.49E+02	3.68E+02	3.46E+02	3.46E+02	3.51E+02	**3.08E+02**
	STD	2.93E+01	1.56E+01	1.22E+01	9.79E+00	7.42E+00	7.24E+00	**7.02E+00**	2.31E+01
	Symbol	=	+	+	+	+	+	+	/
	Rank	2.30	4.70	5.13	7.50	4.47	4.37	5.57	**1.97**
F_{25}	Mean	3.51E+02	4.16E+02	3.96E+02	4.08E+02	3.95E+02	3.94E+02	3.94E+02	**3.42E+02**
	STD	2.45E+01	1.10E+01	1.56E+01	1.09E+01	**8.91E+00**	9.35E+00	9.72E+00	1.64E+01
	Symbol	=	+	+	+	+	+	+	/
	Rank	1.87	7.30	4.83	6.57	4.70	4.50	4.73	**1.50**
F_{26}	Mean	4.64E+02	2.09E+02	4.16E+02	4.46E+02	**2.03E+02**	**2.03E+02**	**2.03E+02**	3.93E+02
	STD	5.18E+01	3.37E+00	5.79E+01	4.42E+01	**6.70E−01**	1.08E+00	8.94E−01	1.27E+01
	Symbol	+	−	+	+	−	−	−	/
	Rank	7.63	4.10	5.93	7.10	**1.93**	2.03	2.13	5.13
F_{27}	Mean	1.91E+03	1.79E+03	1.66E+03	1.92E+03	1.73E+03	1.72E+03	1.69E+03	**1.34E+03**
	STD	2.51E+02	2.92E+02	9.70E+01	**7.14E+01**	7.87E+01	7.98E+01	2.55E+02	1.66E+02
	Symbol	+	+	+	+	+	+	+	/
	Rank	6.30	5.80	3.03	7.10	4.33	4.03	4.23	**1.17**
F_{28}	Mean	1.44E+03	6.56E+02	1.39E+03	1.87E+03	**4.00E+02**	**4.00E+02**	**4.00E+02**	1.22E+03
	STD	1.50E+03	1.25E+02	1.29E+03	1.16E+03	1.92E−04	2.10E−03	**1.86E−04**	1.38E+03
	Symbol	+	=	+	+	+	+	=	/
	Rank	5.53	5.67	6.47	7.37	2.80	**2.60**	2.87	2.70
+/ = /−		22/5/1	22/1/5	25/1/2	25/2/1	17/3/8	19/1/8	17/3/8	/
Avg. (Rank)		5.04	5.63	5.48	7.15	3.43	3.34	3.29	**2.62**

following order (from best to worst): EAPSO, CHCLPSO-III, CHCLPSO-II, CHCLPSO-I, QPSO, AWPSO, EPSO, and HGSPSO. Clearly, EAPSO is the best of all algorithms.

As can be seen from Table 5, EAPSO is superior to the compared algorithms on more than sixty percent of test functions, which are F_2, F_3, F_5, F_6, F_7, F_8, F_9, F_{10}, F_{12}, F_{13}, F_{15}, F_{16}, F_{18}, F_{21}, F_{23}, F_{24}, F_{25}, and F_{27}. In addition, CHCLPSO-III is the best of all algorithms on six test functions, i.e. F_1, F_{11}, F_{14}, F_{17}, F_{19}, and F_{22}. CHCLPSO-I, CHCLPSO-II, and AWPSO outperform the other algorithms on two test functions (i.e. F_5 and F_{28}), one test function (i.e. F_{26}), and two test functions (i.e. F_4 and F_{20}), respectively. Besides, QPSO, EPSO, and HGSPSO cannot get the best solutions on any test functions. From the last column of Table 5, all algorithms can be sorted in the following order (from best to worst): EAPSO, CHCLPSO-III, CHCLPSO-I, CHCLPSO-II, AWPSO, QPSO, EPSO, and HGSPSO. Obviously, EAPSO is the best of all algorithms.

In order to further compare the performance differences of the applied algorithms based on the experimental results, Fig. 3 shows radar graph based on the experimental results from Friedman test obtained by the applied algorithms on CEC 2013 test suite. By looking carefully at Fig. 3, EAPSO is superior to the compared algorithms on each type of test functions with each considered dimension. This fully demonstrates that EAPSO has more excellent global search ability than the compared algorithms.

4.5. Comparison on convergence performance

This section is to compare the convergence performance between EAPSO and the other seven algorithms as presented in Fig. S1, Fig. S2, and Fig. S3. The four figures can be found in the supplementary material.

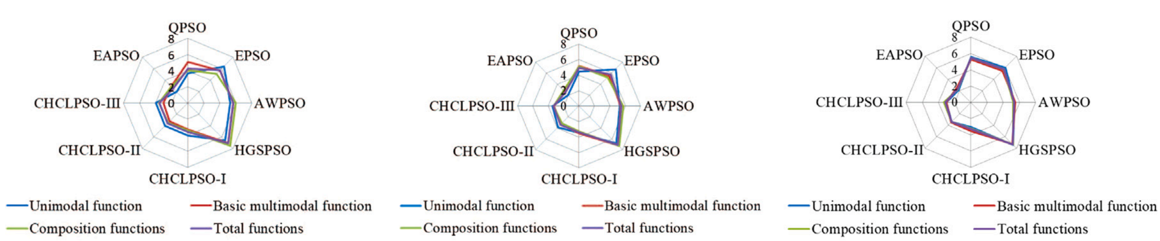
Table 5
Experimental results of EAPSO and the compared algorithms on CEC 2013 test suite with 100-dimensional.

No.	Indicator	QPSO	EPSO	AWPSO	HGSPSO	CHCLPSO-I	CHCLPSO-II	CHCLPSO-III	EAPSO
F_1	Mean	3.46E−02	1.89E−03	6.11E+03	2.54E+04	**2.27E−13**	4.55E−13	**2.27E−13**	2.05E−12
	STD	3.30E−02	7.38E−03	4.17E+02	1.27E+03	6.55E−13	4.92E−13	**4.70E−13**	7.75E−13
	Symbol	+	+	+	+	=	−	−	/
	Rank	5.97	5.03	7.00	8.00	2.18	2.57	**1.27**	3.98
F_2	Mean	1.48E+08	1.24E+08	2.52E+08	6.09E+08	7.75E+07	8.28E+07	8.50E+07	**1.61E+06**
	STD	3.84E+07	2.94E+07	4.06E+07	9.86E+07	1.23E+07	1.31E+07	1.51E+07	**4.05E+05**
	Symbol	+	+	+	+	+	+	+	/
	Rank	5.63	5.07	6.97	8.00	2.87	3.27	3.20	**1.00**
F_3	Mean	3.20E+10	8.41E+10	4.07E+10	1.67E+11	3.17E+10	3.43E+10	3.10E+10	**6.17E+08**
	STD	1.52E+10	2.13E+10	1.37E+10	4.69E+10	7.93E+09	7.90E+09	8.68E+09	**4.16E+08**
	Symbol	+	+	+	+	+	+	+	/
	Rank	3.63	6.93	4.73	7.97	3.87	4.40	3.47	**1.00**
F_4	Mean	8.47E+04	1.16E+05	**1.97E+04**	6.28E+04	6.06E+04	5.86E+04	5.51E+04	3.34E+04
	STD	1.23E+04	1.02E+04	**2.50E+03**	5.79E+03	9.75E+03	6.30E+03	8.22E+03	1.58E+04
	Symbol	+	+	−	+	+	+	+	/
	Rank	6.77	7.97	**1.17**	5.07	4.77	4.23	3.80	2.23
F_5	Mean	1.85E+00	6.56E−04	1.17E+03	4.51E+03	**7.96E−13**	1.02E−12	2.05E−12	1.48E−12
	STD	9.37E−01	7.37E−04	1.39E+02	4.37E+02	5.76E−13	**5.06E−13**	8.26E−13	8.65E−13
	Symbol	+	+	+	+	−	=	=	/
	Rank	6.00	5.00	7.00	8.00	**1.62**	2.43	3.48	2.47
F_6	Mean	2.82E+02	3.69E+02	9.36E+02	2.43E+03	2.69E+02	2.77E+02	2.74E+02	**1.44E+02**
	STD	3.25E+01	4.86E+01	1.01E+02	2.41E+02	2.24E+01	2.62E+01	**2.22E+01**	5.72E+01
	Symbol	+	+	+	+	+	+	+	/
	Rank	3.80	5.83	7.00	8.00	3.17	3.63	3.57	**1.00**
F_7	Mean	1.46E+02	4.41E+02	1.29E+02	2.20E+02	1.69E+02	1.69E+02	1.61E+02	**1.23E+02**
	STD	2.05E+01	3.22E+02	**1.72E+01**	3.55E+01	1.59E+01	1.76E+01	1.80E+01	2.50E+01
	Symbol	+	+	=	+	+	+	+	/
	Rank	3.13	7.90	1.93	6.93	5.03	4.90	4.33	**1.83**
F_8	Mean	**2.13E+01**	**2.13E+01**	**2.13E+01**	**2.13E+01**	**2.13E+01**	**2.13E+01**	**2.13E+01**	**2.13E+01**
	STD	2.34E−02	2.84E−02	**2.09E−02**	2.85E−02	3.83E−02	2.70E−02	3.72E−02	5.01E−02
	Symbol	+	+	+	+	+	+	=	/
	Rank	4.97	5.97	5.13	4.83	3.97	4.63	3.80	**2.70**
F_9	Mean	1.61E+02	1.44E+02	1.20E+02	1.44E+02	1.28E+02	1.29E+02	1.29E+02	**9.73E+01**
	STD	**2.90E+00**	4.23E+00	5.59E+00	4.41E+00	4.28E+00	3.40E+00	4.65E+00	1.15E+01
	Symbol	+	+	+	+	+	+	+	/
	Rank	8.00	6.60	2.30	6.40	3.90	3.83	3.90	**1.07**
F_{10}	Mean	2.23E+02	7.14E+02	1.33E+03	4.24E+03	7.84E+01	8.38E+01	5.93E+01	**7.14E−02**
	STD	8.03E+01	2.29E+02	1.98E+02	4.18E+02	1.49E+01	1.18E+01	1.24E+01	**4.07E−02**
	Symbol	+	+	+	+	+	+	+	/
	Rank	5.00	6.03	6.97	8.00	3.27	3.50	2.23	**1.00**
F_{11}	Mean	6.80E+02	2.76E+01	1.02E+03	1.36E+03	**5.68E−14**	1.14E−13	6.63E−02	2.60E+02
	STD	1.12E+02	1.13E+01	5.15E+01	7.25E+01	1.97E−13	**1.59E−13**	2.52E−01	5.09E+01
	Symbol	+	−	+	+	−	−	−	/
	Rank	6.00	4.00	7.00	8.00	1.83	2.37	**1.80**	5.00
F_{12}	Mean	1.00E+03	1.28E+03	1.04E+03	1.34E+03	9.20E+02	9.52E+02	9.68E+02	**4.18E+02**
	STD	**3.75E+01**	7.22E+01	5.99E+01	6.23E+01	8.15E+01	7.34E+01	7.46E+01	6.83E+01
	Symbol	+	+	+	+	+	+	+	/
	Rank	4.43	7.27	5.27	7.73	3.10	3.47	3.73	**1.00**
F_{13}	Mean	9.86E+02	1.28E+03	1.02E+03	1.37E+03	1.10E+03	1.11E+03	1.08E+03	**7.78E+02**
	STD	4.10E+01	7.85E+01	4.83E+01	9.67E+01	4.10E+01	4.67E+01	**3.67E+01**	1.43E+02
	Symbol	+	+	+	+	+	+	+	/
	Rank	2.37	7.27	3.00	7.73	4.83	5.07	4.57	**1.17**
F_{14}	Mean	3.01E+04	6.92E+02	2.83E+04	3.12E+04	**3.86E+01**	4.32E+01	5.30E+01	9.72E+03
	STD	5.95E+02	2.79E+02	1.08E+03	5.36E+02	**4.60E+01**	6.45E+01	6.65E+01	1.62E+03
	Symbol	+	−	+	+	−	−	−	/
	Rank	7.07	4.00	6.03	7.90	2.40	1.90	**1.70**	5.00
F_{15}	Mean	3.12E+04	1.80E+04	2.97E+04	3.05E+04	1.71E+04	1.72E+04	1.73E+04	**1.44E+04**
	STD	**5.74E+02**	1.44E+03	5.89E+02	5.96E+02	1.11E+03	1.04E+03	8.50E+02	1.33E+03
	Symbol	+	+	+	+	+	+	+	/
	Rank	7.70	3.97	6.20	7.10	3.10	3.17	3.57	**1.20**
F_{16}	Mean	4.17E+00	2.45E+00	4.11E+00	4.12E+00	2.46E+00	2.27E+00	2.29E+00	**7.58E−01**
	STD	2.21E−01	3.94E−01	**1.91E−01**	2.25E−01	2.62E−01	2.90E−01	1.95E−01	3.81E−01
	Symbol	+	+	+	+	+	+	+	/

(*continued on next page*)

Table 5 (*continued*).

No.	Indicator	QPSO	EPSO	AWPSO	HGSPSO	CHCLPSO-I	CHCLPSO-II	CHCLPSO-III	EAPSO
	Rank	7.03	3.83	6.93	7.03	3.97	3.00	3.13	**1.07**
F_{17}	Mean	1.02E+03	4.00E+02	1.32E+03	2.25E+03	1.03E+02	**1.02E+02**	**1.02E+02**	4.82E+02
	STD	6.52E+01	3.87E+01	5.51E+01	9.96E+01	1.30E+00	2.11E−01	**1.52E−01**	7.98E+01
	Symbol	+	−	+	+	−	−	−	/
	Rank	6.00	4.13	7.00	8.00	2.77	1.97	**1.27**	4.87
F_{18}	Mean	1.12E+03	1.74E+03	1.33E+03	2.21E+03	1.15E+03	1.14E+03	1.09E+03	**5.26E+02**
	STD	3.99E+01	1.03E+02	**3.60E+01**	9.70E+01	5.45E+01	5.04E+01	5.40E+01	8.07E+01
	Symbol	+	+	+	+	+	+	+	/
	Rank	3.27	7.00	6.00	8.00	4.10	3.87	2.77	**1.00**
F_{19}	Mean	8.38E+01	2.73E+01	1.61E+02	6.31E+03	5.09E+00	4.43E+00	**3.18E+00**	3.71E+01
	STD	7.27E+00	6.50E+00	2.17E+01	2.08E+03	9.75E−01	6.55E−01	**4.59E−01**	7.80E+00
	Symbol	+	+	+	+	−	−	−	/
	Rank	6.00	4.23	7.00	8.00	2.73	2.23	**1.03**	4.77
F_{20}	Mean	**5.00E+01**	**5.00E+01**	**5.00E+01**	**5.00E+01**	**5.00E+01**	**5.00E+01**	**5.00E+01**	**5.00E+01**
	STD	1.54E−11	1.06E−13	1.26E−03	**0.00E+00**	**0.00E+00**	7.72E−12	8.51E−10	**0.00E+00**
	Symbol	=	=	=	=	=	=	=	/
	Rank	4.53	4.55	**4.10**	4.67	4.67	4.43	4.38	4.67
F_{21}	Mean	6.55E+02	7.35E+02	1.15E+03	3.25E+03	4.22E+02	4.24E+02	4.28E+02	**3.56E+02**
	STD	4.44E+01	4.45E+02	3.90E+02	7.80E+02	**3.15E+00**	1.04E+01	2.95E+01	6.14E+01
	Symbol	+	+	+	+	+	+	+	/
	Rank	5.40	5.57	6.97	7.97	2.87	3.07	3.13	**1.03**
F_{22}	Mean	2.96E+04	5.62E+02	3.13E+04	3.29E+04	9.38E+01	9.17E+01	**8.31E+01**	1.29E+04
	STD	7.70E+02	2.23E+02	1.32E+03	1.01E+03	**6.14E+01**	7.48E+01	7.80E+01	2.42E+03
	Symbol	+	−	+	+	−	−	−	/
	Rank	6.10	3.97	7.07	7.83	2.27	2.10	**1.67**	5.00
F_{23}	Mean	3.12E+04	2.48E+04	3.22E+04	3.23E+04	2.30E+04	2.31E+04	2.30E+04	**1.99E+04**
	STD	**7.24E+02**	2.12E+03	9.87E+02	9.81E+02	1.20E+03	1.46E+03	1.32E+03	2.80E+03
	Symbol	+	+	+	+	+	+	+	/
	Rank	6.30	4.20	7.33	7.37	2.93	3.13	3.13	**1.60**
F_{24}	Mean	5.22E+02	5.79E+02	5.44E+02	5.94E+02	5.53E+02	5.52E+02	5.44E+02	**4.44E+02**
	STD	5.28E+01	2.38E+01	1.74E+01	1.61E+01	1.36E+01	**1.30E+01**	1.93E+01	2.38E+01
	Symbol	+	+	+	+	+	+	+	/
	Rank	3.60	6.63	4.00	7.57	4.43	4.77	3.93	**1.07**
F_{25}	Mean	5.97E+02	7.43E+02	6.41E+02	6.94E+02	6.56E+02	6.55E+02	6.59E+02	**5.22E+02**
	STD	5.37E+01	2.52E+01	2.20E+01	2.17E+01	1.53E+01	1.56E+01	**1.40E+01**	2.69E+01
	Symbol	+	+	+	+	+	+	+	/
	Rank	2.33	7.93	3.67	6.77	4.57	4.67	4.93	**1.13**
F_{26}	Mean	6.97E+02	4.68E+02	6.05E+02	6.67E+02	2.96E+02	**2.40E+02**	**2.40E+02**	5.33E+02
	STD	1.27E+01	2.26E+02	1.47E+01	**9.36E+00**	1.72E+02	1.06E+02	1.08E+02	2.50E+01
	Symbol	+	=	+	+	−	−	−	/
	Rank	7.90	4.83	5.13	6.87	2.70	**2.17**	2.30	4.10
F_{27}	Mean	4.30E+03	4.09E+03	3.60E+03	4.18E+03	3.67E+03	3.73E+03	3.88E+03	**2.84E+03**
	STD	2.15E+02	1.87E+02	1.74E+02	**1.39E+02**	6.29E+02	6.39E+02	8.90E+01	2.01E+02
	Symbol	+	+	+	+	+	+	+	/
	Rank	7.47	6.17	2.63	6.80	3.33	3.93	4.60	**1.07**
F_{28}	Mean	4.73E+03	6.09E+03	4.43E+03	6.73E+03	**3.76E+03**	3.91E+03	3.90E+03	3.95E+03
	STD	9.02E+02	1.11E+03	9.78E+02	9.21E+02	**6.77E+02**	7.41E+02	8.49E+02	1.17E+03
	Symbol	+	+	=	+	=	=	=	/
	Rank	4.70	6.83	4.77	7.67	**2.87**	3.03	3.03	3.10
+/ = /−		27/1/0	22/2/4	24/3/1	27/1/0	18/3/7	18/3/7	17/4/7	/
Avg. (Rank)		5.40	5.67	5.37	7.29	3.36	3.42	3.13	**2.36**



(a) 30-dimensional (b) 50-dimensional (c) 100-dimensional

Fig. 3. Radar graph based on the experimental results from Friedman test obtained by the applied algorithms on CEC 2013 test suite.

Table 6
The sensitivity of the dimension obtained by six algorithms on 28 test functions (normalized processing).

No.	QPSO	EPSO	AWPSO	HGSPSO	CHCLPSO-I	CHCLPSO-II	CHCLPSO-III	EAPSO
F_1	9.42E−7	1.59E−4	1.96E−1	8.04E−1	**0.00E+0**	6.19E−18	**0.00E+0**	4.96E−17
F_2	9.56E−2	1.05E−1	1.84E−1	4.43E−1	5.46E−2	5.81E−2	5.87E−2	**1.22E−3**
F_3	7.68E−2	2.15E−1	9.30E−2	3.83E−1	7.46E−2	8.33E−2	7.35E−2	**1.36E−3**
F_4	2.07E−1	2.62E−1	**3.87E−2**	1.56E−1	9.11E−2	8.56E−2	7.12E−2	8.84E−2
F_5	2.65E−4	2.44E−3	1.87E−1	8.10E−1	**1.22E−16**	1.79E−16	3.01E−16	2.19E−16
F_6	5.11E−2	7.01E−2	1.79E−1	4.97E−1	5.47E−2	5.62E−2	5.41E−2	**3.79E−2**
F_7	1.17E−1	2.99E−1	6.50E−2	1.53E−1	1.06E−1	1.01E−1	1.03E−1	**5.51E−2**
F_8	1.24E−1	**1.16E−1**	1.26E−1	1.28E−1	1.24E−1	1.27E−1	1.27E−1	1.29E−1
F_9	1.53E−1	1.34E−1	1.12E−1	1.34E−1	1.24E−1	1.24E−1	1.24E−1	**9.54E−2**
F_{10}	2.89E−2	1.55E−1	1.85E−1	5.98E−1	1.19E−2	1.23E−2	8.86E−3	**8.71E−6**
F_{11}	2.05E−1	4.55E−3	3.06E−1	4.01E−1	**0.00E+0**	1.52E−17	1.78E−5	8.36E−2
F_{12}	1.24E−1	1.59E−1	1.30E−1	1.67E−1	1.18E−1	1.21E−1	1.28E−1	**5.26E−2**
F_{13}	1.10E−1	1.43E−1	1.15E−1	1.51E−1	1.30E−1	1.32E−1	1.28E−1	**8.98E−2**
F_{14}	3.03E−1	1.35E−3	2.84E−1	3.15E−1	**2.88E−4**	4.11E−4	3.70E−4	9.52E−2
F_{15}	1.74E−1	9.79E−2	1.72E−1	1.73E−1	9.98E−2	1.02E−1	1.02E−1	**7.98E−2**
F_{16}	1.93E−1	5.53E−2	2.06E−1	1.93E−1	1.15E−1	1.01E−1	1.05E−1	**3.23E−2**
F_{17}	1.79E−1	6.04E−2	2.32E−1	3.96E−1	1.64E−2	**1.62E−2**	**1.62E−2**	8.30E−2
F_{18}	1.08E−1	1.67E−1	1.29E−1	2.15E−1	1.12E−1	1.11E−1	1.09E−1	**4.88E−2**
F_{19}	1.32E−2	1.87E−3	2.58E−2	9.52E−1	7.30E−4	6.18E−4	**3.81E−4**	5.82E−3
F_{20}	1.28E−1	1.28E−1	1.24E−1	**1.23E−1**	1.25E−1	1.24E−1	**1.23E−1**	1.24E−1
F_{21}	1.18E−1	1.10E−1	1.83E−1	4.16E−1	2.68E−2	3.29E−2	**2.63E−2**	8.68E−2
F_{22}	2.70E−1	1.50E−3	2.96E−1	3.04E−1	**1.01E−3**	1.27E−3	1.06E−3	1.24E−1
F_{23}	1.46E−1	1.14E−1	1.55E−1	1.52E−1	1.12E−1	1.13E−1	1.15E−1	**9.35E−2**
F_{24}	1.15E−1	1.41E−1	1.24E−1	1.42E−1	1.32E−1	1.29E−1	1.29E−1	**8.76E−2**
F_{25}	1.16E−1	1.55E−1	1.24E−1	1.34E−1	1.28E−1	1.28E−1	1.28E−1	**8.77E−2**
F_{26}	3.00E−1	1.09E−1	1.74E−1	2.07E−1	3.88E−2	**1.64E−2**	1.67E−2	1.39E−1
F_{27}	1.49E−1	1.37E−1	1.15E−1	1.35E−1	1.24E−1	1.28E−1	1.25E−1	**8.72E−2**
F_{28}	1.65E−1	1.39E−1	1.14E−1	1.54E−1	**9.74E−2**	1.01E−1	1.01E−1	1.28E−1
Avg.	1.35E−1	1.10E−1	1.56E−1	3.16E−1	7.21E−2	7.16E−2	7.04E−2	**6.92E−2**

Fig. S1 compares the convergence performance among the applied algorithms on CEC 2013 test suite with 30-dimensional. From Fig. S1, the selected functions include three unimodal functions (i.e. F_2, F_3, and F_5), nine basic multimodal functions (i.e. F_6, F_8, F_9, F_{10}, F_{12}, F_{13}, F_{15}, F_{16}, and F_{18}), and three composition functions (i.e. F_{23}, F_{24}, and F_{27}). Looking at Fig. S1, EAPSO shows better convergence performance than the compared algorithms on the selected functions. Further, EAPSO has slight advantages over the compared algorithms on F_5, F_6, and F_{10}. EAPSO has obvious advantages over the compared algorithms on F_2, F_3, F_8, F_9, F_{12}, F_{13}, F_{15}, F_{16}, F_{18}, F_{23}, F_{24}, and F_{27}.

Fig. S2 displays the differences of convergence performance between EAPSO and the compared algorithms on CEC 2013 test suite with 50-dimensional. As can be seen from Fig. S2, the selected functions are two unimodal functions (i.e. F_2 and F_3), nine basic multimodal functions (i.e. F_6, F_8, F_9, F_{10}, F_{12}, F_{13}, F_{15}, F_{16}, and F_{18}), and four composition functions (i.e. F_{23}, F_{24}, F_{25}, and F_{27}). As shown in Fig. S2, EAPSO can get the better solutions with the faster speed than the compared algorithms. In addition, EAPSO has obvious advantages over the compared algorithms on about eighty-six percent of the selected functions including F_2, F_3, F_8, F_9, F_{12}, F_{13}, F_{15}, F_{16}, F_{18}, F_{23}, F_{24}, F_{25}, and F_{27}.

Fig. S3 presents the comparisons of the convergence performance between EAPSO and the rest seven algorithms on CEC 2013 test suite with 100-dimensional. Looking at Fig. S3, the selected functions are two unimodal functions (i.e. F_2 and F_3), ten basic multimodal functions (i.e. F_6, F_8, F_9, F_{10}, F_{12}, F_{13}, F_{15}, F_{16}, F_{18}, and F_{19}), and six composition functions (i.e. F_{21}, F_{23}, F_{24}, F_{25}, F_{27}, and F_{28}). As can be seen from Fig. S3, EAPSO can offer better convergence performance than the compared algorithms on the selected 18 test functions. Moreover, EAPSO shows clear advantages over the compared algorithms on 14 of 18 test functions, which include F_2, F_3, F_8, F_9, F_{12}, F_{13}, F_{15}, F_{16}, F_{18}, F_{23}, F_{24}, F_{25}, F_{27}, and F_{28}.

Based on the above discussion, Fig. S1, Fig. S2, and Fig. S3 show the convergence curves of 15 test functions with 30-dimensional, 15 test functions with 50-dimensional, and 18 test functions with 100-dimensional, respectively. In other words, in terms of convergence performance, EAPSO is superior to the compared algorithms on more than half of test functions from CEC 2013 test suite with each considered dimension, which supports the excellent convergence performance of EAPSO.

4.6. Comparison on the impact of dimension

This section is to compare the sensitivity of EAPSO and the compared algorithms on dimension. Note that, as the number of dimensions increases of one problem, it is increasingly difficult to solve this problem. Thus, the sensitivity of an algorithm for the dimension is vital for the quality of the obtained solution. To achieve this, the sensitivity of the dimension is defined as follows:

$$S=\frac{1}{n-1}\sum_{i=1}^{n-1}\left|\frac{V_{\mathrm{mean},i+1}-V_{\mathrm{mean},i}}{D_{i+1}-D_i}\right|, i=1,2,3, \tag{14}$$

where S is the sensitivity of the dimension, n is the number of the considered dimensions, D_i is the value of the ith dimension, and $V_{\mathrm{mean},i}$ is the mean value on the ith dimension. The designed experiments consider three dimensions, i.e. 30, 50, and 100. Thus, n is equal to 3, D_1 is equal to 30, D_2 is equal to 50, and D_3 is equal to 100. According to Eq. (14), a smaller S denotes a lower sensitivity of the dimension, which proves the stronger stable of an algorithm.

The results of the sensitivity of the dimension obtained by EAPSO and the compared algorithms on 30 test functions are presented in Table 6. The best results in Table 6 are in bold. As can be seen from Table 6, EAPSO can get the optimal the sensitivity of the dimension on 15 test functions including F_2, F_3, F_6, F_7, F_9, F_{10}, F_{12}, F_{13}, F_{15}, F_{16}, F_{18}, F_{23}, F_{24}, F_{25}, and F_{27}. In addition, EPSO, AWPSO, HGSPSO, CHCLPSO-I, CHCLPSO-II, and CHCLPSO-III can achieve the optimal the sensitivity of the dimension on one test functions (i.e. F_8), one test functions (i.e. F_4), one test function (i.e. F_{20}), six test functions (i.e. F_1, F_5, F_{11}, F_{14}, F_{22}, and F_{28}), two test function (i.e. F_{17} and F_{26}), and five test functions (i.e. F_1, F_{17}, F_{19}, F_{20}, and F_{21}), respectively. From the last column of Table 6, all algorithms can be sorted in the following order (from best to worst): EAPSO, CHCLPSO-III, CHCLPSO-II, CHCLPSO-I, EPSO, QPSO, AWPSO, and HGSPSO. Clearly, EAPSO has the smallest sensitivity of the dimension, which is more stable than the compared algorithms and has more potential to be applied to high-dimensional optimization problems.

Table 7
The experimental results obtained by EAPSO with different population sizes.

No.	Index	30-dimensional			50-dimensional			100-dimensional		
		EAPSO$_{N=20}$	EAPSO$_{N=50}$	EAPSO$_{N=100}$	EAPSO$_{N=20}$	EAPSO$_{N=50}$	EAPSO$_{N=100}$	EAPSO$_{N=20}$	EAPSO$_{N=50}$	EAPSO$_{N=100}$
F_1	Mean	3.18E−12	2.27E−13	**2.27E−13**	4.87E−11	1.14E−12	**2.27E−13**	2.16E−06	8.41E−12	**2.05E−12**
	STD	2.63E−12	6.31E−13	**2.31E−13**	1.02E−10	6.06E−13	**4.96E−13**	1.08E−05	6.95E−12	**7.75E−13**
F_2	Mean	1.05E+05	**1.01E+05**	1.74E+05	4.68E+05	**3.81E+05**	5.25E+05	1.72E+06	**1.22E+06**	1.61E+06
	STD	4.34E+04	**3.85E+04**	6.78E+04	**1.44E+05**	1.93E+05	1.65E+05	4.63E+05	**3.45E+05**	4.05E+05
F_3	Mean	2.39E+08	1.15E+08	**2.28E+07**	5.32E+08	2.96E+08	**8.84E+07**	1.19E+10	1.36E+09	**6.17E+08**
	STD	2.23E+08	2.05E+08	**2.63E+07**	3.33E+08	5.13E+08	**8.08E+07**	7.59E+09	1.10E+09	**4.16E+08**
F_4	Mean	1.25E+04	1.42E+04	**8.58E+03**	**1.74E+04**	2.53E+04	1.93E+04	6.59E+04	5.44E+04	**3.34E+04**
	STD	6.21E+03	1.10E+04	**4.81E+03**	**1.10E+04**	1.48E+04	1.25E+04	2.39E+04	2.32E+04	**1.58E+04**
F_5	Mean	3.75E−12	2.27E−13	**1.14E−13**	1.06E−09	1.14E−12	**2.27E−13**	8.75E−07	8.41E−12	**1.48E−12**
	STD	5.13E−12	3.85E−13	**1.98E−13**	5.63E−09	5.92E−13	**3.76E−13**	3.93E−06	9.20E−12	**8.65E−13**
F_6	Mean	1.06E+01	8.32E+00	**7.31E+00**	4.36E+01	4.37E+01	**4.33E+01**	1.64E+02	1.51E+02	**1.44E+02**
	STD	**9.39E+00**	1.73E+01	1.72E+01	1.29E+01	**9.81E+00**	1.08E+01	**5.14E+01**	4.92E+01	5.72E+01
F_7	Mean	1.20E+02	7.41E+01	**6.99E+01**	1.36E+02	1.12E+02	**8.65E+01**	1.40E+04	1.64E+02	**1.23E+02**
	STD	3.32E+01	**2.56E+01**	2.59E+01	2.40E+01	2.35E+01	**2.08E+01**	2.60E+04	4.09E+01	**2.50E+01**
F_8	Mean	2.10E+01	**2.09E+01**	**2.09E+01**	**2.11E+01**	**2.11E+01**	**2.11E+01**	**2.13E+01**	**2.13E+01**	**2.13E+01**
	STD	**6.26E−02**	7.45E−02	6.76E−02	6.09E−02	7.15E−02	**5.16E−02**	**4.85E−02**	4.98E−02	5.01E−02
F_9	Mean	2.86E+01	2.15E+01	**1.71E+01**	5.83E+01	4.35E+01	**3.72E+01**	1.36E+02	1.11E+02	**9.73E+01**
	STD	**2.97E+00**	4.39E+00	3.09E+00	**4.35E+00**	5.00E+00	5.37E+00	**6.29E+00**	1.04E+01	1.15E+01
F_{10}	Mean	9.98E−02	6.62E−02	**3.18E−02**	7.81E−02	6.99E−02	**5.47E−02**	**5.00E−02**	7.06E−02	7.14E−02
	STD	6.44E−02	3.16E−02	**1.83E−02**	4.83E−02	4.21E−02	**3.28E−02**	**3.32E−02**	4.03E−02	4.07E−02
F_{11}	Mean	6.95E+01	4.29E+01	**3.52E+01**	2.17E+02	1.13E+02	**9.35E+01**	8.58E+02	3.64E+02	**2.60E+02**
	STD	3.68E+01	1.72E+01	**1.40E+01**	5.06E+01	3.32E+01	**2.60E+01**	1.82E+02	7.49E+01	**5.09E+01**
F_{12}	Mean	1.67E+02	1.04E+02	**6.94E+01**	3.77E+02	2.07E+02	**1.55E+02**	1.14E+03	5.87E+02	**4.18E+02**
	STD	5.69E+01	2.78E+01	**2.71E+01**	9.80E+01	4.91E+01	**2.97E+01**	2.58E+02	9.90E+01	**6.83E+01**
F_{13}	Mean	2.55E+02	1.85E+02	**1.41E+02**	6.25E+02	3.99E+02	**3.13E+02**	1.66E+03	9.95E+02	**7.78E+02**
	STD	5.20E+01	4.23E+01	**3.70E+01**	1.01E+02	5.92E+01	**5.65E+01**	1.82E+02	**1.26E+02**	1.43E+02
F_{14}	Mean	1.77E+03	**1.44E+03**	1.84E+03	4.19E+03	3.68E+03	**3.37E+03**	1.21E+04	1.01E+04	**9.72E+03**
	STD	8.88E+02	**3.32E+02**	6.29E+02	1.15E+03	9.51E+02	**9.39E+02**	**1.01E+03**	1.83E+03	1.62E+03
F_{15}	Mean	4.16E+03	4.08E+03	**3.97E+03**	8.22E+03	7.71E+03	**7.38E+03**	1.66E+04	1.55E+04	**1.44E+04**
	STD	7.14E+02	**6.59E+02**	6.76E+02	**6.96E+02**	1.09E+03	1.02E+03	1.70E+03	1.38E+03	**1.33E+03**
F_{16}	Mean	1.03E+00	7.51E−01	**4.64E−01**	1.60E+00	9.30E−01	**5.73E−01**	2.22E+00	1.28E+00	**7.58E−01**
	STD	4.01E−01	4.50E−01	**3.89E−01**	5.39E−01	5.66E−01	**3.44E−01**	6.12E−01	6.56E−01	**3.81E−01**
F_{17}	Mean	1.37E+02	8.13E+01	**7.55E+01**	4.14E+02	2.07E+02	**1.52E+02**	1.66E+03	7.44E+02	**4.82E+02**
	STD	5.07E+01	1.47E+01	**1.14E+01**	1.28E+02	3.35E+01	**2.70E+01**	2.51E+02	9.83E+01	**7.98E+01**
F_{18}	Mean	2.46E+02	1.23E+02	**9.19E+01**	5.90E+02	2.60E+02	**1.74E+02**	2.29E+03	9.19E+02	**5.26E+02**
	STD	8.31E+01	2.34E+01	**1.69E+01**	1.58E+02	4.20E+01	**3.07E+01**	3.81E+02	1.47E+02	**8.07E+01**
F_{19}	Mean	1.65E+01	5.02E+00	**3.52E+00**	6.32E+01	1.80E+01	**7.99E+00**	8.58E+02	8.71E+01	**3.71E+01**
	STD	8.97E+00	2.25E+00	**1.31E+00**	3.17E+01	5.98E+00	**2.14E+00**	4.64E+02	2.44E+01	**7.80E+00**
F_{20}	Mean	1.49E+01	1.47E+01	**1.45E+01**	2.46E+01	**2.42E+01**	2.44E+01	**5.00E+01**	**5.00E+01**	**5.00E+01**
	STD	**3.66E−01**	8.05E−01	1.04E+00	**7.36E−01**	1.28E+00	1.33E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
F_{21}	Mean	2.82E+02	**2.74E+02**	2.91E+02	8.34E+02	8.45E+02	**7.46E+02**	3.58E+02	**3.56E+02**	**3.56E+02**
	STD	1.04E+02	**7.92E+01**	8.30E+01	3.81E+02	3.85E+02	**3.37E+02**	6.79E+01	**6.14E+01**	**6.14E+01**
F_{22}	Mean	2.60E+03	2.81E+03	**2.45E+03**	5.92E+03	5.91E+03	**5.22E+03**	1.65E+04	1.60E+04	**1.29E+04**
	STD	6.13E+02	**5.18E+02**	5.98E+02	**1.05E+03**	1.06E+03	1.23E+03	3.09E+03	3.19E+03	**2.42E+03**
F_{23}	Mean	5.28E+03	**4.72E+03**	4.80E+03	9.92E+03	9.34E+03	**9.04E+03**	2.23E+04	2.07E+04	**1.99E+04**
	STD	9.16E+02	**8.54E+02**	9.04E+02	**1.35E+03**	1.43E+03	1.42E+03	2.35E+03	**2.16E+03**	2.80E+03
F_{24}	Mean	2.85E+02	2.65E+02	**2.55E+02**	3.62E+02	3.25E+02	**3.08E+02**	5.82E+02	5.01E+02	**4.44E+02**
	STD	1.08E+01	1.23E+01	**1.02E+01**	1.37E+01	**1.24E+01**	2.31E+01	**2.02E+01**	2.14E+01	2.38E+01
F_{25}	Mean	3.02E+02	2.85E+02	**2.75E+02**	3.94E+02	3.60E+02	**3.42E+02**	6.63E+02	5.86E+02	**5.22E+02**
	STD	1.14E+01	1.30E+01	**1.11E+01**	**1.40E+01**	1.69E+01	1.64E+01	2.92E+01	**2.68E+01**	2.69E+01
F_{26}	Mean	3.30E+02	3.19E+02	**3.09E+02**	4.12E+02	**3.89E+02**	3.93E+02	6.46E+02	5.83E+02	**5.33E+02**
	STD	7.88E+01	7.14E+01	**6.72E+01**	8.55E+01	7.64E+01	**1.27E+01**	2.06E+01	2.60E+01	**2.50E+01**
F_{27}	Mean	1.11E+03	9.16E+02	**8.42E+02**	1.83E+03	1.50E+03	**1.34E+03**	4.01E+03	3.25E+03	**2.84E+03**
	STD	1.11E+02	1.21E+02	**8.77E+01**	**1.39E+02**	1.49E+02	1.66E+02	2.06E+02	2.47E+02	**2.01E+02**
F_{28}	Mean	6.92E+02	**3.38E+02**	4.11E+02	1.95E+03	**1.14E+03**	1.22E+03	1.12E+04	5.23E+03	**3.95E+03**
	STD	6.79E+02	**2.08E+02**	3.38E+02	1.93E+03	**1.37E+03**	1.38E+03	2.63E+03	1.49E+03	**1.17E+03**

4.7. Comparison on the impact of population size

This section is to discuss the impact of population size on the global search ability of EAPSO. Experimental results of EAPSO with three different population sizes have been shown in Table 7. In this table, $\text{EAPSO}_{N=20}$, $\text{EAPSO}_{N=50}$, and $\text{EAPSO}_{N=100}$ are EAPSO with $N = 20$, EAPSO with $N = 50$, EAPSO with $N = 100$, respectively. The best results in Table 7 are in bold.

According to the experimental results presented in Table 7, $\text{EAPSO}_{N=20}$ can achieve or share the best MEAN on two test functions (i.e. F_4 and F_8) with 50-dimensional and three test functions (i.e. F_8, F_{10}, and F_{20}) with 100-dimensional. In addition, $\text{EAPSO}_{N=50}$ can offer or share the MEAN on five test functions (i.e. F_2, F_8, F_{14}, F_{21}, and F_{28}) with 30-dimensional, five test functions (i.e. F_2, F_8, F_{20}, F_{26}, and F_{28}) with 50-dimensional, and four test functions (i.e. F_2, F_8, F_{20}, and F_{21}) with 100-dimensional. Besides, in terms of MEAN, $\text{EAPSO}_{N=100}$ can find or share the best solutions on 23 test functions (i.e. F_1, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}, F_{11}, F_{12}, F_{13}, F_{15}, F_{16}, F_{17}, F_{18}, F_{19}, F_{20}, F_{22}, F_{23}, F_{24}, F_{25}, F_{26}, and F_{27}) with 30-dimensional, 23 test functions (i.e. F_1, F_3, F_5, F_6, F_7, F_8, F_9, F_{10}, F_{11}, F_{12}, F_{13}, F_{14}, F_{15}, F_{16}, F_{17}, F_{18}, F_{19}, F_{21}, F_{22}, F_{23}, F_{24}, F_{25}, and F_{27}) with 50-dimensional, and 26 test functions (i.e. F_1, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{11}, F_{12}, F_{13}, F_{14}, F_{15}, F_{16}, F_{17}, F_{18}, F_{19}, F_{20}, F_{21}, F_{22}, F_{23}, F_{24}, F_{25}, F_{26}, F_{27}, and F_{28}) with 100-dimensional.

Based on the above discussion, in terms of solution accuracy, $\text{EAPSO}_{N=100}$ is remarkably better than $\text{EAPSO}_{N=20}$ and $\text{EAPSO}_{N=50}$ on each considered dimension. This can be explained as follows. There are two remarkable features in EAPSO: (1) only common particles are used for performing the search task and outstanding particles go directly into the next generation population in one loop; (2) the built learning mechanism consists of six candidate learning strategies. Thus, if population size is too small, the designed learning strategies cannot be fully used and the advantages of EAPSO are not be shown. Motivated by this, when EAPSO is applied to solve an unknown problem, the suggested population size is 100. When the suggested population size cannot find promising solution, it is a good attempt to increase properly the population size.

5. EAPSO on engineering design problems

This section is to evaluate the optimization performance of EAPSO on constrained engineering design problems. The excellent global search ability of EAPSO on unconstrained numerical problems has been proven in Section 4, which provides a good foundation for achieving the goal of this section. Note that, the selected three constrained engineering design problems including speed reducer design problem, tension/compression spring design problem and rolling element bearing design problem, which are typical multimodal problems. In addition, as done in [48], penalty functions are introduced to address the constrained conditions of the three engineering problems. Further, using penalty functions is aimed at transforming a constrained optimization problem into an unconstrained optimization problem with the help of the given penalty terms. The detailed description of the penalty functions can be found in [48]. Besides, for EAPSO, population size is set to 100 as suggested in Section 4.7; the maximum number of function evaluations is set to 15,000; 30 times independently runs are executed on each engineering design problem. To better show the competitiveness of EAPSO in solving the considered engineering problems, the solutions of EAPSO are compared with those of the reported methods as presented in Tables 8, 9, and 10. Note that, the compared solutions are extracted directly from the corresponding references.

5.1. Speed reducer design problem

Speed reducer problem is a very classical engineering design problem, which is to minimize the weight of a speed reducer. This problem has seven design variables including the face width $b(x_1)$, the module of teeth $m(x_2)$, the number of teeth in the pinion $z(x_3)$, the length of the first shaft between bearings $l_1(x_4)$, the length of the second shaft between bearings $l_2(x_5)$, the diameter of the first shaft $d_1(x_6)$, and the diameter of the second shaft $d_2(x_7)$. In addition, this problem needs to address 11 constraints consisting of seven nonlinear constraints and four linear constraints. The mathematical model of this problem can be represented by

$$\begin{aligned}
&\operatorname{Min} f(\boldsymbol{x}) = f\left(x_1, x_2, x_3, x_4, x_5, x_6, x_7\right) = \\
&0.7854 x_1 x_2^2\left(3.3333 x_3^2 + 14.9334 x_3 - 43.0934\right) \\
&-1.508 x_1\left(x_6^2 + x_7^2\right) \\
&+7.4777\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4 x_6^2 + x_5 x_7^2\right) \\
&\text{Subject to:} \\
&g_1(\boldsymbol{x}) = \frac{27}{x_1 x_2^2 x_3} - 1 \le 0 \\
&g_2(\boldsymbol{x}) = \frac{397.5}{x_1 x_2^2 x_3} - 1 \le 0 \\
&g_3(\boldsymbol{x}) = \frac{1.93 x_4^3}{x_2 x_6^4 x_3} - 1 \le 0 \\
&g_4(\boldsymbol{x}) = \frac{1.93 x_5^3}{x_2 x_7^4 x_3} - 1 \le 0 \\
&g_5(\boldsymbol{x}) = \frac{\left(\left(\frac{745 x_4}{x_2 x_3}\right)^2 + 16.9\times 10^6\right)^{1/2}}{110 x_6^3} - 1 \le 0 \\
&g_6(\boldsymbol{x}) = \frac{\left(\left(\frac{745 x_5}{x_2 x_3}\right)^2 + 157.5\times 10^6\right)^{1/2}}{85 x_7^3} - 1 \le 0 \\
&g_7(\boldsymbol{x}) = \frac{x_2 x_3}{40} - 1 \le 0 \\
&g_8(\boldsymbol{x}) = \frac{5 x_2}{x_1} - 1 \le 0 \\
&g_9(\boldsymbol{x}) = \frac{x_1}{12 x_2} - 1 \le 0 \\
&g_{10}(\boldsymbol{x}) = \frac{1.5 x_6 + 1.9}{x_4} - 1 \le 0 \\
&g_{11}(\boldsymbol{x}) = \frac{1.1 x_7 + 1.9}{x_5} - 1 \le 0
\end{aligned}, \tag{15}$$

where $2.6 \le x_1 \le 3.6, 0.7 \le x_2 \le 0.8, 17 \le x_3 \le 28, 7.3 \le x_4, x_5 \le 8.3, 2.9 \le x_6 \le 3.9, 5.0 \le x_7 \le 5.5$.

Table 8 presents the optimal solutions of EAPSO and 10 compared metaheuristic algorithms including IGPSO [49], LSHADE [50], GSSA [51], CSA [52], ASO [53], ASOINU [53], HHSC [54], CPA [55], AO [56], and GBO [57] in solving speed reducer design problem. As can be seen from Table 8, the solution of IGPSO is the best, which is 2994.381. In addition, EAPSO shows strong competitiveness, whose solution is 2994.471066 that is very close to that of IGPSO. Besides, the solutions of the other nine algorithms, i.e. LSHADE, GSSA, CSA, ASO, ASOINU, HHSC, CPA, AO, and GBO, cannot compete with those of IGPSO and EAPSO. Note that, the solutions of AO and ASO are more than 3000, which are far more than those of the other ten algorithms. In other words, AO and ASO are not suitable for solving the speed reducer design problem.

5.2. Tension/compression spring design problem

Pressure vessel design problem is often employed to verify the optimization ability of a population-based optimization algorithm, which is aimed at minimizing the weight of a tension/compression spring. This problem has three design variables including the wire diameter

Table 8
The best solutions of EAPSO and the reported methods on the speed reducer design problem.

Algorithm	The optimal variable							Optimal weight
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
IGPSO	3.49999	0.700000	17.00000	7.30000	7.71516	3.350214	5.286517	2994.381
LSHADE	3.500000	0.700000	17.00000	7.30000	7.80000	3.350215	5.286683	2996.348165
GSSA	3.500957	0.7	17	7.331317	7.806692	3.351851	5.28669	2997.5658
CSA	3.5	0.7	17	7.3	7.8	3.35021467	5.286668323	2996.348165
ASO	3.500000	0.700000	17.0755	7.30000	8.1198	3.4366	5.2935	3051.31905
ASOINU	3.500000	0.700000	17.00000	7.30000	7.80000	3.3502	5.2865	2996.24478
HHSC	3.50379	0.7	17.0	7.3	7.7294014	3.356511	5.28668965	2997.89844
CPA	3.500000	0.700000	17.00000	7.30000	7.80000	3.35021478	5.286668325	2996.21851300
AO	3.5021	0.7000	17.000	7.3099	7.7476	3.3641	5.2994	3007.7328
GBO	3.4999	0.70	17.00	7.30	7.80	3.3502	5.2866	2996.3481
EAPSO	3.500000	0.700000	17.00000	7.30000	7.7153199	3.35021467	5.28665446	2994.471066

Table 9
The best solutions of EAPSO and the reported methods on the tension/compression spring design problem.

Algorithm	The optimal variable			Optimal weight
	x1	x2	x3	
PCFMO	0.052405	0.37416	10.338	0.012867
UAPSO	0.051712	0.357264	11.257028	0.012665
HPSO	0.05	0.317425	14.02777	0.012719
LSFQPSO	0.051513	0.352394	11.552000	0.0126724
SCSO	0.0500	0.3175	14.0200	0.012717
IGWO	0.051701	0.356983	11.2756	0.012667
GJO	0.0515793	0.354055	11.4484	0.01266752
BWO	0.0517	0.3568	11.3132	0.012703
ICHIMP-SHO	0.051324	0.347614	11.86041	0.0126915
DO	0.051215	0.345416	11.983708	0.012669
EAPSO	0.0515219	0.3527102	11.527838	0.012666

x_1, the mean coil diameter x_2, and the number of active x_3 [58]. In addition, four constraints need to be conceded, which include one linear constraint and three nonlinear constraints. The mathematical model of this problem can be expressed by

$$\begin{aligned}
&\mathrm{Min} f(\boldsymbol{x}) = f\left(x_1, x_2, x_3\right) = \left(x_3+2\right) x_2 x_1^2 \\
&\text{Subject to:} \\
&g_1(\boldsymbol{x}) = 1 - \frac{x_2^3 x_3}{71{,}785 x_1^4} \le 0 \\
&g_2(\boldsymbol{x}) = 4x_2^2 - \frac{x_1 x_2}{12.566\left(x_2 x_1^3 - x_1^4\right)} + \frac{1}{5108 x_1^2} - 1 \le 0 \quad , \\
&g_3(\boldsymbol{x}) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \le 0 \\
&g_4(\boldsymbol{x}) = x_2 + \frac{x_1}{1.5} - 1 \le 0
\end{aligned} \tag{16}$$

where $0.05 \le x_1 \le 2, 0.25 \le x_2 \le 1.30, 2.00 \le x_3 \le 15.00$.

The optimal solutions of EAPSO and 10 compared metaheuristic algorithms including PCFMO [59], UAPSO [60], HPSO [61], LSFQPSO [62], SCSO [63], IGWO [64], GJO [65], BWO [66], DO [67], and ICHIMP-SHO [68] on speed reducer design problem have been displayed in Table 9. According to Table 9, UAPSO can get the best solution, i.e. 0.012665. In addition, EAPSO can obtain a very competitive solution, i.e. 0.012666. IGWO and GJO also show good global search ability, whose solutions are very similar with those of UAPSO and EAPSO. Besides, the solutions of HPSO, SCSO, and BWO are more than 0.0127, which are significantly more than those of the other eight algorithms. That is, HPSO, SCSO, and BWO have no advantages over the other eight algorithms in solving this problem.

5.3. Rolling element bearing design problem

Rolling element bearing design problem is a very challenging engineering problem, whose objective is to maximize the dynamic load carting capacity of a rolling element bearing. This problem has 10 design variables consisting of the pitch diameter $D_m(x_1)$, the ball diameter $D_b(x_2)$, the number of balls $Z(x_2)$, the inner raceway curvature coefficients $f_i(x_4)$, the outer raceway curvature coefficients $f_o(x_5)$, the minimum ball diameter limiter $K_{Dmin}(x_6)$, the maximum ball diameter limiter $K_{Dmin}(x_7)$, the parameter for outer ring strength consideration $\varepsilon(x_8)$, the parameter for mobility condition $e(x_9)$, and the bearing width limiter $\zeta(x_{10})$. In addition, nine constraints need to be addressed. The mathematical model of this problem can be defined by

$$\begin{aligned}
&\mathrm{Max} f(\mathbf{x}) = f(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}) \\
&= \begin{cases} f_c Z^{2/3} D_b^{1.8}, \text{if } D \le 25.4 \text{ mm} \\ 3.647 f_c Z^{2/3} D_b^{1.4}, \text{if } D > 25.4 \text{ mm} \end{cases} \\
&\text{Subject to} \\
&g_1(\mathbf{x}) = \frac{\phi_0}{2\sin^{-1}(D_b/D_m)} - Z + 1 \ge 0 \\
&g_2(\mathbf{x}) = 2D_b - K_{D\min}(D-d) \ge 0 \\
&g_3(\mathbf{x}) = K_{D\max}(D-d) - 2D_b \ge 0 \quad , \\
&g_4(\mathbf{x}) = \zeta B_w - D_b \le 0 \\
&g_5(\mathbf{x}) = D_m - 0.5(D+d) \ge 0 \\
&g_6(\mathbf{x}) = (0.5+e)(D+d) - D_m \ge 0 \\
&g_7(\mathbf{x}) = 0.5\left(D - D_m - D_b\right) - \varepsilon D_b \ge 0 \\
&g_8(\mathbf{x}) = f_i \ge 0.515 \\
&g_9(\mathbf{x}) = f_o \ge 0.515
\end{aligned} \tag{17}$$

where $\gamma = \frac{D_b}{D_m}$, $f_i = \frac{r_i}{D_b}$, $f_o = \frac{r_o}{D_b}$, $T = D - d - 2D_b$, $D = 160$, $d = 90$, $B_w = 30$, $D = 160$, $r_i = r_o = 11.033$, $0.5(D+d) \le D_m \le 0.6(D+d)$, $0.15(D-d) \le D_b \le 0.45(D-d)$, $4 \le Z \le 50, 0, .515 \le f_i \le 0.6$, $0.515 \le f_o \le 0.6$, $0.4 \le K_{D\min} \le 0.5$, $0.6 \le K_{D\max} \le 0.7$, $0.3 \le e \le 0.4$, $0.02 \le \varepsilon \le 0.1$, $0.6 \le \zeta \le 0.85$. In addition, Eq. (S3) and Eq. (S4) describe the computing methods of f_c and ϕ_0, which can be found in the supplementary material.

At present, many researchers have solved the rolling element bearing design problem by some powerful metaheuristic algorithms, such as SMA-PS [69], HHO [70], PVS [71], EPO [72], ESA [73], ARO [74], and AFT [75]. Table 10 presents the obtained best solutions from EAPSO and the compared algorithms in solving the rolling element bearing design problem. According to Table 10, EASPO can find the best solution, i.e. 85 549.239146. In addition, ARO can achieve the competitive solution, which are 85 548.5106. Besides, the solutions of SMA-PS and PVS are very bad, which cannot compete with those of the rest algorithms.

6. Discussion for the effectiveness of the improved strategies

This section is to discuss the effectiveness of the improved strategies based on the experimental results in Sections 4 and 5.

The improved strategies in EAPSO can be stated as follows: three types of elite archives are built, which are used for generating six learning strategies to enhance the ability of EAPSO to escape from the local optimal solutions. As can be seen from Section 3, the improved strategies in EAPSO have clearly hierarchical property, which can be stated as follows.

Table 10
The best solutions of EAPSO and the reported methods on the rolling element bearing design problem.

Variable	SMA-PS	HHO	PVS	EPO	ARO	ESA	AFT	EAPSO
x_1	125.7234255	125	125.719060	125	125.7189	125	125	125.7190556
x_2	21.42285806	21	21.425590	21.41890	21	21.41750	21.418	21.42559024
x_3	11.00138214	11.092073	11.000000	10.94113	10.5403	10.94109	11.356	11
x_4	0.515	0.515	0.515000	0.515	0.5150	0.515	0.515	0.515
x_5	0.515	0.515	0.515000	0.515	0.5150	0.515	0.515	0.515
x_6	0.492067898	0.4	0.400430	0.4	0.4459	0.4	0.4	0.5
x_7	0.7	0.6	0.680160	0.7	0.672132	0.7	0.680	0.6435792
x_8	0.3	0.3	0.300000	0.3	0.3000	0.3	0.3	0.3
x_9	0.030435006	0.050474	0.079990	0.02	0.0825	0.02	0.02	0.04721879
x_{10}	0.638555041	0.6	0.700000	0.6	0.6317	0.6	0.622	0.6053521
Optimal load	81 859.74120	83 011.88329	81 859.74121	85 067.983.	85 548.5106	85 070.085	85 206.641	85 549.239146

- The built three types of elite archives have obviously hierarchical property. Individuals in the archive $\boldsymbol{A}^t$ are extracted directly from the historical best solutions of outstanding particles and updated before starting one loop. The individuals of archive $\boldsymbol{B}^t$ are from the newborn promising particles during the execution of one loop. For PSO, one global best solution will be obtained after completing one loop. The individuals of archive $\boldsymbol{C}^t$ are from the obtained global best solutions. According to the quality of the individuals, the three archives can be sorted in the following order (from high to low): archive $\boldsymbol{C}^t$, archive $\boldsymbol{B}^t$, and archive $\boldsymbol{A}^t$.
- The designed six learning strategies have obviously hierarchical property. On one hand, each case has three different individuals to guide the search direction of common particle as shown in Eqs. (5)–(10). An individual is the global best solution $\boldsymbol{g}^t$ and the another individual is the best of $\boldsymbol{a}_m{}^t$, $\boldsymbol{b}_r{}^t$, and $\boldsymbol{c}_q{}^t$ in the case 1, case 2, and case 3, respectively. It is clear that case 1, case 2, and case 3 have significant advantages over case 4, case 5, and case 6 in terms of the selected individuals to guide the search direction of common particle i. On the other hand, as shown in Algorithm 3, if the fitness value of a common particle outperforms the mean fitness value of all common particles, this common particle will choose case 1, case 2, or case 3 to update its velocity, which is very helpful for accelerating the convergence of the population; otherwise, it will use case 4, case 5, or case 6 to update its velocity, which is beneficial for keeping population diversity. Thus, the communication among particles can be significantly improved.

Sections 4 and 5 check the performance of EAPSO by numerical optimization problems and engineering optimization problems, respectively. In Section 4, EAPSO and the seven recently reported variants are used for solving 28 test functions from CEC 2013 test suite with dimensions 30–100. As shown in Tables 3, 4, and 5, EAPSO can archive or share the best results on more than half of test functions with each considered dimension. In addition, according to the experimental results of Wilcoxon signed-rank test and Friedman test presented in Tables 3, 4, and 5, EAPSO clearly outperforms the other seven algorithms. EAPSO is the best of all algorithms based on the ranking results shown in Tables 3, 4, and 5. Besides, the experimental results presented in Table 6 prove that EAPSO is much less sensitive to the dimension of the problem compared with the other seven algorithms. That is, EAPSO has obvious advantages over the compared algorithms in solving large scale optimization problems. Fig. S1, Fig. S2, and Fig. S3 prove the convergence advantages of EAPSO over the compared algorithms. In Section 5, three well-known engineering design problems, i.e. speed reducer design problem, tension/compression spring design problem, and rolling element bearing design problem are solved by EAPSO. According to the best solutions of EAPSO and the reported algorithms shown in Tables 8, 9, and 10, the solutions of EAPSO are very competitive and EAPSO has great potential to be used for solving complex practical engineering problems. Note that, more than 82 percent of numerical problems and the applied three engineering design problems are typical multimodal optimization problems. Experimental results support the excellent global search ability of EAPSO on multimodal problems.

Based on the above discussion, the obtained experimental results prove the effectiveness of the improved strategies in EAPSO. The improved strategies can effectively enhance the ability of EAPSO to avoid premature convergence.

7. Conclusion

Given simple structure and powerful search ability, PSO is always a hot topic since it proposed. PSO has two control parameters including global acceleration coefficient and local acceleration coefficient. In addition, PSO-w is a very popular improved version of PSO, which introduces another control parameter called inertia weight. Note that, the three control parameters, i.e. inertia weight, global acceleration coefficient and local acceleration coefficient, are used in almost all the reported variants of PSO. Different optimization problems usually have different features. Thus, a common phenomenon is that the values of the three control parameters in these variants are not exactly the same for solving different optimization problems. To get the optimal solution of an unknown problem, how to set the most suitable values of the three control parameters for these variants of PSO is a huge challenge. Therefore, how to avoid this challenge is a very interesting and valuable research topic.

This work presents a novel variant of PSO called EAPSO for solving complex multimodal problems. The most remarkably feature of EAPSO is that it only needs the essential population size and terminal condition and does not refer to any other control parameters in solving optimization problems. The core of EAPSO is the built three types of elite archives to be used for saving some promising particles with three hierarchical levels. Based on the built three levels of archives, six candidate learning strategies are designed. Note that, the designed six candidate learning strategies also can be divided into two different hierarchical levels. In other words, the designed learning mechanism has not only rich learning strategies, but a hierarchical structure.

To investigate the effectiveness of the improved strategies introduced to EAPSO, EAPSO and seven powerful variants of PSO are employed for solving CEC 2013 test suite with dimensions 30–100. Experimental results show that EAPSO is significantly superior to the compared variants of PSO in terms of solution accuracy, significance test results produced by Wilcoxon signed-rank test, ranking results produced by Friedman test, convergence performance, and the sensitivity of the dimension. EAPSO is also used to solve three constrained engineering design problems, i.e. speed reducer design problem, tension/compression spring design problem and rolling element bearing design problem. EAPSO can get very competitive solutions. This fully proves the excellent global search ability of EAPSO for solving complex multimodal problems.

Further work will focus on the following three aspects. Firstly, EAPSO is our first attempt to design PSO without the control parameters. The success of EAPSO motivates us to develop more powerful variants of PSO without any control parameters. Then, given the excellent global search ability of EAPSO, we intend to apply EAPSO to solve more practical engineering problems, such as flexible job shop scheduling with outsourcing operations and job priority constraints, image classification, and photovoltaic solar cell/module parameter extraction. Lastly, the core of EAPSO is the created three types of elite archives. Note that, in one loop, extracting and updating the elements from the three archives reduce the computation efficiency of EAPSO. Thus, how to improve the computation efficiency of EAPSO is our another interest.

CRediT authorship contribution statement

Yiying Zhang: Conceptualization, Methodology, Original draft, Writing – review & editing, Software.

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.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by the Research Foundation for Talented Scholars of Jiangsu University, China (Grant No. 21JDG065) and the Natural Science Foundation of Jiangsu Province, China (Grant No. BK20220509).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.swevo.2022.101212.

References

[1] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, Vol. 4, 1995, pp. 1942–1948, http://dx.doi.org/10.1109/ICNN.1995.488968, vol.4.

[2] F.E. Fernandes Junior, G.G. Yen, Particle swarm optimization of deep neural networks architectures for image classification, Swarm Evol. Comput. 49 (2019) 62–74, http://dx.doi.org/10.1016/j.swevo.2019.05.010.

[3] X. Fu, F.T. Chan, B. Niu, N.S. Chung, T. Qu, A three-level particle swarm optimization with variable neighbourhood search algorithm for the production scheduling problem with mould maintenance, Swarm Evol. Comput. 50 (2019) 100572, http://dx.doi.org/10.1016/j.swevo.2019.100572.

[4] Z. Wang, J. Zhang, S. Yang, An improved particle swarm optimization algorithm for dynamic job shop scheduling problems with random job arrivals, Swarm Evol. Comput. 51 (2019) 100594, http://dx.doi.org/10.1016/j.swevo.2019.100594.

[5] A. Darwish, D. Ezzat, A.E. Hassanien, An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis, Swarm Evol. Comput. 52 (2020) 100616, http://dx.doi.org/10.1016/j.swevo.2019.100616.

[6] F. Han, W.-T. Chen, Q.-H. Ling, H. Han, Multi-objective particle swarm optimization with adaptive strategies for feature selection, Swarm Evol. Comput. 62 (2021) 100847, http://dx.doi.org/10.1016/j.swevo.2021.100847.

[7] Z. Cui, J. Zhang, D. Wu, X. Cai, H. Wang, W. Zhang, J. Chen, Hybrid many-objective particle swarm optimization algorithm for green coal production problem, Inform. Sci. 518 (2020) 256–271, http://dx.doi.org/10.1016/j.ins.2020.01.018.

[8] Z.-J. Wang, Z.-H. Zhan, W.-J. Yu, Y. Lin, J. Zhang, T.-L. Gu, J. Zhang, Dynamic group learning distributed particle swarm optimization for large-scale optimization and its application in cloud workflow scheduling, IEEE Trans. Cybern. 50 (6) (2020) 2715–2729, http://dx.doi.org/10.1109/TCYB.2019.2933499.

[9] J. Liang, A. Qin, P. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295, http://dx.doi.org/10.1109/TEVC.2005.857610.

[10] J. Lu, J. Zhang, J. Sheng, Enhanced multi-swarm cooperative particle swarm optimizer, Swarm Evol. Comput. 69 (2022) 100989, http://dx.doi.org/10.1016/j.swevo.2021.100989.

[11] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360), 1998, pp. 69–73, http://dx.doi.org/10.1109/ICEC.1998.699146.

[12] B. Biswal, P.K. Dash, B.K. Panigrahi, Power quality disturbance classification using fuzzy C-means algorithm and adaptive particle swarm optimization, IEEE Trans. Ind. Electron. 56 (1) (2009) 212–220, http://dx.doi.org/10.1109/TIE.2008.928111.

[13] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm Evol. Comput. 24 (2015) 11–24, http://dx.doi.org/10.1016/j.swevo.2015.05.002.

[14] X. Xia, L. Gui, F. Yu, H. Wu, B. Wei, Y.-L. Zhang, Z.-H. Zhan, Triple archives particle swarm optimization, IEEE Trans. Cybern. 50 (12) (2020) 4862–4875, http://dx.doi.org/10.1109/TCYB.2019.2943928.

[15] Y. Zhang, X. Liu, F. Bao, J. Chi, C. Zhang, P. Liu, Particle swarm optimization with adaptive learning strategy, Knowl.-Based Syst. 196 (2020) 105789, http://dx.doi.org/10.1016/j.knosys.2020.105789.

[16] L. Wang, H. Geng, P. Liu, K. Lu, J. Kolodziej, R. Ranjan, A.Y. Zomaya, Particle swarm optimization based dictionary learning for remote sensing big data, Knowl.-Based Syst. 79 (2015) 43–50, http://dx.doi.org/10.1016/j.knosys.2014.10.004.

[17] G. Xu, Q. Cui, X. Shi, H. Ge, Z.-H. Zhan, H.P. Lee, Y. Liang, R. Tai, C. Wu, Particle swarm optimization based on dimensional learning strategy, Swarm Evol. Comput. 45 (2019) 33–51, http://dx.doi.org/10.1016/j.swevo.2018.12.009.

[18] P. Ghosh, H. Zafar, S. Das, A. Abraham, Hierarchical dynamic neighborhood based particle swarm optimization for global optimization, in: 2011 IEEE Congress of Evolutionary Computation (CEC), 2011, pp. 757–764, http://dx.doi.org/10.1109/CEC.2011.5949695.

[19] L. Pan, Y. Zhao, L. Li, Neighborhood-based particle swarm optimization with discrete crossover for nonlinear equation systems, Swarm Evol. Comput. 69 (2022) 101019, http://dx.doi.org/10.1016/j.swevo.2021.101019.

[20] L. Zhen, Y. Liu, W. Dongsheng, Z. Wei, Parameter estimation of software reliability model and prediction based on hybrid wolf pack algorithm and particle swarm optimization, IEEE Access 8 (2020) 29354–29369, http://dx.doi.org/10.1109/ACCESS.2020.2972826.

[21] H. bin Ouyang, L. qun Gao, X. yong Kong, S. Li, D. xuan Zou, Hybrid harmony search particle swarm optimization with global dimension selection, Inform. Sci. 346–347 (2016) 318–337, http://dx.doi.org/10.1016/j.ins.2016.02.007.

[22] M.A. Rodriguez-Guerrero, A.Y. Jaen-Cuellar, R.D. Carranza-Lopez-Padilla, R.A. Osornio-Rios, G. Herrera-Ruiz, R.d.J. Romero-Troncoso, Hybrid approach based on GA and PSO for parameter estimation of a full power quality disturbance parameterized model, IEEE Trans. Ind. Inform. 14 (3) (2018) 1016–1028, http://dx.doi.org/10.1109/TII.2017.2743762.

[23] P. Civicioglu, Backtracking search optimization algorithm for numerical optimization problems, Appl. Math. Comput. 219 (15) (2013) 8121–8144, http://dx.doi.org/10.1016/j.amc.2013.02.017.

[24] Z. Zeng, M. Zhang, T. Chen, Z. Hong, A new selection operator for differential evolution algorithm, Knowl.-Based Syst. 226 (2021) 107150, http://dx.doi.org/10.1016/j.knosys.2021.107150.

[25] B.H. Abed-alguni, N.A. Alawad, M. Barhoush, R. Hammad, Exploratory cuckoo search for solving single-objective optimization problems, Soft Comput. 25 (15) (2021) 10167–10180.

[26] H. Han, X. Bai, H. Han, Y. Hou, J. Qiao, Self-adjusting multitask particle swarm optimization, IEEE Trans. Evol. Comput. 26 (1) (2022) 145–158, http://dx.doi.org/10.1109/TEVC.2021.3098523.

[27] S.-C. Wang, Y.-H. Liu, A PSO-based fuzzy-controlled searching for the optimal charge pattern of Li-ion batteries, IEEE Trans. Ind. Electron. 62 (5) (2015) 2983–2993, http://dx.doi.org/10.1109/TIE.2014.2363049.

[28] K. Ishaque, Z. Salam, M. Amjad, S. Mekhilef, An improved particle swarm optimization (PSO)–based MPPT for PV with reduced steady-state oscillation, IEEE Trans. Power Electron. 27 (8) (2012) 3627–3638, http://dx.doi.org/10.1109/TPEL.2012.2185713.

[29] Z.-H. Liu, H.-L. Wei, Q.-C. Zhong, K. Liu, X.-S. Xiao, L.-H. Wu, Parameter estimation for VSI-fed PMSM based on a dynamic PSO with learning strategies, IEEE Trans. Power Electron. 32 (4) (2017) 3154–3165, http://dx.doi.org/10.1109/TPEL.2016.2572186.

[30] H. Ge, L. Sun, G. Tan, Z. Chen, C.L.P. Chen, Cooperative hierarchical PSO with two stage variable interaction reconstruction for large scale optimization, IEEE Trans. Cybern. 47 (9) (2017) 2809–2823, http://dx.doi.org/10.1109/TCYB.2017.2685944.

[31] H. Li, D. Yang, W. Su, J. Lü, X. Yu, An overall distribution particle swarm optimization MPPT algorithm for photovoltaic system under partial shading, IEEE Trans. Ind. Electron. 66 (1) (2019) 265–275, http://dx.doi.org/10.1109/TIE.2018.2829668.

[32] C. Sun, Y. Jin, R. Cheng, J. Ding, J. Zeng, Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems, IEEE Trans. Evol. Comput. 21 (4) (2017) 644–660, http://dx.doi.org/10.1109/TEVC.2017.2675628.

[33] X. Li, X. Yao, Cooperatively coevolving particle swarms for large scale optimization, IEEE Trans. Evol. Comput. 16 (2) (2012) 210–224, http://dx.doi.org/10.1109/TEVC.2011.2112662.

[34] H. Rakhshani, A. Rahati, Snap-drift cuckoo search: A novel cuckoo search optimization algorithm, Appl. Soft Comput. 52 (2017) 771–794, http://dx.doi.org/10.1016/j.asoc.2016.09.048.
[35] M.Z. Ali, N.H. Awad, P.N. Suganthan, R.G. Reynolds, A modified cultural algorithm with a balanced performance for the differential evolution frameworks, Knowl.-Based Syst. 111 (2016) 73–86, http://dx.doi.org/10.1016/j.knosys.2016.08.005.
[36] Y. Wang, B. Li, T. Weise, Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems, Inform. Sci. 180 (12) (2010) 2405–2420, http://dx.doi.org/10.1016/j.ins.2010.02.015.
[37] M.Z. Ali, N.H. Awad, P.N. Suganthan, Multi-population differential evolution with balanced ensemble of mutation strategies for large-scale global optimization, Appl. Soft Comput. 33 (2015) 304–327, http://dx.doi.org/10.1016/j.asoc.2015.04.019.
[38] M. Lozano, D. Molina, F. Herrera, Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems, Soft Comput. 15 (11) (2011) 2085–2087, http://dx.doi.org/10.1007/s00500-010-0639-2.
[39] X. Zhang, S.Y. Yuen, A directional mutation operator for differential evolution algorithms, Appl. Soft Comput. 30 (2015) 529–548, http://dx.doi.org/10.1016/j.asoc.2015.02.005.
[40] L. Cui, G. Li, X. Wang, Q. Lin, J. Chen, N. Lu, J. Lu, A ranking-based adaptive artificial bee colony algorithm for global numerical optimization, Inform. Sci. 417 (2017) 169–185, http://dx.doi.org/10.1016/j.ins.2017.07.011.
[41] Z. Liang, K. Hu, Q. Zhu, Z. Zhu, An enhanced artificial bee colony algorithm with adaptive differential operators, Appl. Soft Comput. 58 (2017) 480–494, http://dx.doi.org/10.1016/j.asoc.2017.05.005.
[42] J. Sun, W. Xu, B. Feng, A global search strategy of quantum-behaved particle swarm optimization, in: IEEE Conference on Cybernetics and Intelligent Systems, 2004, Vol. 1, 2004, pp. 111–116, http://dx.doi.org/10.1109/ICCIS.2004.1460396, vol.1.
[43] F. Li, X. Cai, L. Gao, Ensemble of surrogates assisted particle swarm optimization of medium scale expensive problems, Appl. Soft Comput. 74 (2019) 291–305, http://dx.doi.org/10.1016/j.asoc.2018.10.037.
[44] T.A. Khan, S.H. Ling, A novel hybrid gravitational search particle swarm optimization algorithm, Eng. Appl. Artif. Intell. 102 (2021) 104263, http://dx.doi.org/10.1016/j.engappai.2021.104263.
[45] D. Yousri, D. Allam, M. Eteiba, P.N. Suganthan, Chaotic heterogeneous comprehensive learning particle swarm optimizer variants for permanent magnet synchronous motor models parameters estimation, Iran. J. Sci. Technol. Trans. Electr. Eng. 44 (3) (2020) 1299–1318.
[46] W. Liu, Z. Wang, Y. Yuan, N. Zeng, K. Hone, X. Liu, A novel sigmoid-function-based adaptive weighted particle swarm optimizer, IEEE Trans. Cybern. 51 (2) (2021) 1085–1093, http://dx.doi.org/10.1109/TCYB.2019.2925015.
[47] C.T. Gray, Introduction to quality engineering: Designing quality into products and processes, G. Taguchi, Asian productivity organization, 1986. Number of pages: 191. price: $29 (U.K.), Qual. Reliab. Eng. Int. 4 (2) (1988) 198, http://dx.doi.org/10.1002/qre.4680040216, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/qre.4680040216.
[48] A.H. Gandomi, X.-S. Yang, A.H. Alavi, Mixed variable structural optimization using firefly algorithm, Comput. Struct. 89 (23) (2011) 2325–2336, http://dx.doi.org/10.1016/j.compstruc.2011.08.002.
[49] H. bin Ouyang, L. qun Gao, S. Li, X. yong Kong, Improved global-best-guided particle swarm optimization with learning operation for global optimization problems, Appl. Soft Comput. 52 (2017) 987–1008, http://dx.doi.org/10.1016/j.asoc.2016.09.030.
[50] S. Aras, E. Gedikli, H.T. Kahraman, A novel stochastic fractal search algorithm with fitness-distance balance for global numerical optimization, Swarm Evol. Comput. 61 (2021) 100821, http://dx.doi.org/10.1016/j.swevo.2020.100821.
[51] B. Nautiyal, R. Prakash, V. Vimal, G. Liang, H. Chen, Improved salp swarm algorithm with mutation schemes for solving global optimization and engineering problems, Eng. Comput. (2021) http://dx.doi.org/10.1007/s00366-020-01252-z.
[52] Z. kai Feng, W. jing Niu, S. Liu, Cooperation search algorithm: A novel metaheuristic evolutionary intelligence algorithm for numerical optimization and engineering optimization problems, Appl. Soft Comput. 98 (2021) 106734, http://dx.doi.org/10.1016/j.asoc.2020.106734.
[53] P. Sun, H. Liu, Y. Zhang, L. Tu, Q. Meng, An intensify atom search optimization for engineering design problems, Appl. Math. Model. 89 (2021) 837–859, http://dx.doi.org/10.1016/j.apm.2020.07.052.
[54] L. Abualigah, A. Diabat, M. Altalhi, M.A. Elaziz, Improved gradual change-based harris hawks optimization for real-world engineering design problems, Eng. Comput. (2022) http://dx.doi.org/10.1007/s00366-021-01571-9.
[55] K.M. Ong, P. Ong, C.K. Sia, A carnivorous plant algorithm for solving global optimization problems, Appl. Soft Comput. 98 (2021) 106833, http://dx.doi.org/10.1016/j.asoc.2020.106833.
[56] L. Abualigah, D. Yousri, M. Abd Elaziz, A.A. Ewees, M.A. Al-qaness, A.H. Gandomi, Aquila optimizer: A novel meta-heuristic optimization algorithm, Comput. Ind. Eng. 157 (2021) 107250, http://dx.doi.org/10.1016/j.cie.2021.107250.
[57] I. Ahmadianfar, O. Bozorg-Haddad, X. Chu, Gradient-based optimizer: A new metaheuristic optimization algorithm, Inform. Sci. 540 (2020) 131–159, http://dx.doi.org/10.1016/j.ins.2020.06.037.
[58] S. Gupta, K. Deep, A novel random walk grey wolf optimizer, Swarm Evol. Comput. 44 (2019) 101–112, http://dx.doi.org/10.1016/j.swevo.2018.01.001.
[59] S.-C. Chu, X.-W. Xu, S.-Y. Yang, J.-S. Pan, Parallel fish migration optimization with compact technology based on memory principle for wireless sensor networks, Knowl.-Based Syst. 241 (2022) 108124, http://dx.doi.org/10.1016/j.knosys.2022.108124.
[60] M. Isiet, M. Gadala, Self-adapting control parameters in particle swarm optimization, Appl. Soft Comput. 83 (2019) 105653, http://dx.doi.org/10.1016/j.asoc.2019.105653.
[61] H. Liu, X. Zhang, H. Liang, L. Tu, Stability analysis of the human behavior-based particle swarm optimization without stagnation assumption, Expert Syst. Appl. 159 (2020) 113638, http://dx.doi.org/10.1016/j.eswa.2020.113638.
[62] X. Liu, G.-G. Wang, L. Wang, LSFQPSO: quantum particle swarm optimization with optimal guided Lévy flight and straight flight for solving optimization problems, Eng. Comput. (2021) http://dx.doi.org/10.1007/s00366-021-01497-2.
[63] A. Seyyedabbasi, F. Kiani, Sand cat swarm optimization: a nature-inspired algorithm to solve global optimization problems, Eng. Comput. (2022) http://dx.doi.org/10.1007/s00366-022-01604-x.
[64] W. Long, J. Jiao, X. Liang, M. Tang, Inspired grey wolf optimizer for solving large-scale function optimization problems, Appl. Math. Model. 60 (2018) 112–126, http://dx.doi.org/10.1016/j.apm.2018.03.005.
[65] N. Chopra, M. Mohsin Ansari, Golden jackal optimization: A novel nature-inspired optimizer for engineering applications, Expert Syst. Appl. 198 (2022) 116924, http://dx.doi.org/10.1016/j.eswa.2022.116924.
[66] C. Zhong, G. Li, Z. Meng, Beluga whale optimization: A novel nature-inspired metaheuristic algorithm, Knowl.-Based Syst. 251 (2022) 109215, http://dx.doi.org/10.1016/j.knosys.2022.109215.
[67] S. Zhao, T. Zhang, S. Ma, M. Chen, Dandelion optimizer: A nature-inspired metaheuristic algorithm for engineering applications, Eng. Appl. Artif. Intell. 114 (2022) 105075, http://dx.doi.org/10.1016/j.engappai.2022.105075.
[68] C. Kumari, V.K. Kamboj, S. Bath, S.L. Tripathi, M. Khatri, S. Sehgal, et al., A boosted chimp optimizer for numerical and engineering design optimization challenges, Eng. Comput. (2022) http://dx.doi.org/10.1007/s00366-021-01591-5.
[69] A. Bala Krishna, S. Saxena, V.K. Kamboj, hSMA-PS: a novel memetic approach for numerical and engineering design challenges, Eng. Comput. 38 (4) (2022) 3513–3547, http://dx.doi.org/10.1007/s00366-021-01371-1.
[70] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872, http://dx.doi.org/10.1016/j.future.2019.02.028.
[71] P. Savsani, V. Savsani, Passing vehicle search (PVS): A novel metaheuristic algorithm, Appl. Math. Model. 40 (5) (2016) 3951–3978, http://dx.doi.org/10.1016/j.apm.2015.10.040.
[72] G. Dhiman, V. Kumar, Emperor penguin optimizer: A bio-inspired algorithm for engineering problems, Knowl.-Based Syst. 159 (2018) 20–50, http://dx.doi.org/10.1016/j.knosys.2018.06.001.
[73] G. Dhiman, ESA: a hybrid bio-inspired metaheuristic optimization approach for engineering problems, Eng. Comput. 37 (1) (2021) 323–353, http://dx.doi.org/10.1007/s00366-019-00826-w.
[74] L. Wang, Q. Cao, Z. Zhang, S. Mirjalili, W. Zhao, Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems, Eng. Appl. Artif. Intell. 114 (2022) 105082, http://dx.doi.org/10.1016/j.engappai.2022.105082.
[75] M. Braik, M.H. Ryalat, H. Al-Zoubi, A novel meta-heuristic algorithm for solving numerical optimization problems: Ali Baba and the forty thieves, Neural Comput. Appl. 34 (1) (2022) 409–455, http://dx.doi.org/10.1007/s00521-021-06392-x.