

RESEARCH

Open Access



Particle swarm optimization using multi-information characteristics of all personal-best information

Song Huang¹, Na Tian³, Yan Wang^{1*} and Zhicheng Ji^{1,2}

*Correspondence:
wangyan88@jiangnan.edu.cn
¹ School of Internet of Things
Engineering, Jiangnan
University, 1800 Lihu Avenue,
Wuxi 214122, Jiangsu
Province, China
Full list of author information
is available at the end of the
article

Abstract

Convergence stagnation is the chief difficulty to solve hard optimization problems for most particle swarm optimization variants. To address this issue, a novel particle swarm optimization using multi-information characteristics of all personal-best information is developed in our research. In the modified algorithm, two positions are defined by personal-best positions and an improved cognition term with three positions of all personal-best information is used in velocity update equation to enhance the search capability. This strategy could make particles fly to a better direction by discovering useful information from all the personal-best positions. The validity of the proposed algorithm is assessed on twenty benchmark problems including unimodal, multi-modal, rotated and shifted functions, and the results are compared with that obtained by some published variants of particle swarm optimization in the literature. Computational results demonstrate that the proposed algorithm finds several global optimum and high-quality solutions in most case with a fast convergence speed.

Keywords: Premature convergence, Intelligence algorithm, Particle swarm optimization, Personal-best position

Background

Particle swarm optimization (PSO) is a bio-inspired optimization algorithm introduced by Eberhart and Kennedy (1995), and is enlightened by the interaction and communication of bird flocking or fish schooling. PSO has attracted a great deal of attention as a treatment for high-dimensional nonlinear optimization problem due to its better computational efficiency and simple implementation. With the development of intelligent manufacturing and complex system, many engineering problems are becoming increasingly complex to optimize, and thus time-consuming computation and premature convergence often occurs in complicated optimization process. Therefore, many PSO variants with new techniques have been proposed to address the above problems.

Some researchers got insight into three control parameters, named after acceleration coefficients and inertia weight, to develop PSO variants (Beielstein et al. 2002; Zhang et al. 2014; Shi and Eberhart 1998a, b). In (Shi and Eberhart 1998b), linearly decreasing inertia weight particle swarm optimization (LPSO) was developed by modified inertia weight and the introduction of this dynamic inertia weight highly strengthened the

performance of PSO algorithm. In recent research, multiple swarms or multiple layers strategies had already been proved to be an effective strategy to improve the performance of PSO (Sun and Li 2014; Yadav and Deep 2014; Lim and Isa 2014a; Wang et al. 2014). Sun and Li presented a cooperative particle swarm optimization (TCPSTO) with two-swarm (the slave swarm and the master swarm) for optimization problem in large scale search space (Sun and Li 2014) and two subswarms using shrinking hypersphere PSO (SHPSO) and DE were also used in new co-swarm PSO for constrained optimization problems (Yadav and Deep 2014). Multiple layers strategies, such as adaptive two-layer particle swarm optimization algorithm with elitist learning strategy (ATLPSO-ELS) (Lim and Isa 2014a) and multi-layer particle swarm optimization (MLPSO) (Wang et al. 2014), were also used to solve complex problems. PSO with different topologies has different exploration/exploitation ability and performance (Bonyadi et al. 2014; Lim and Isa 2014b, c). Many new topology strategies [time-adaptive topology (Bonyadi et al. 2014), adaptive time-varying topology connectivity (Lim and Isa 2014b), increasing topology connectivity (Lim and Isa 2014c)] were also applied to PSO. Comparing with fully-connected topology or regular topology, these topologies could lead to a different optimization process. In recent years, new techniques such as Levy flight (Haklı and Uğuz 2014), parallel cell coordinate system (Hu and Yen 2015), competitive and cooperative (Li et al. 2015) and orthogonal design (Qin et al. 2015) had also been adopted in PSO.

Many learning strategies are introduced to PSO to enhance the adaptability for complex optimization problems as learning behavior stemming from social animals plays a key role in animals' adaptation to the changing environment (Cheng and Jin 2015; Rao and Patel 2013; Lim and Isa 2014d; e; Shi and Eberhart 1999). Cheng and Jin presented a modified particle swarm optimization using social learning mechanism (SL-PSO) (Cheng and Jin 2015) and some concept of teachers, tutorial training and self motivated learning was proposed in teaching-learning-based PSO by Rao and Patel for performance enhancements (Rao and Patel 2013). Using teaching and peer-learning behaviors, a bidirectional teaching and peer-learning PSO (BTPLPSO) (Lim and Isa 2014d) and a two learning phases PSO (TPLPSO) (Lim and Isa 2014e) were proposed by Lim and Isa simultaneously.

Communication and learning behavior is a distinguishing feature among social animals and it improves social efficiency. Sharing information mechanism plays a key role in this behavior. To share personal-best information fairness, a particle swarm optimizer using several multi-information characteristics of all personal-best information is developed in this paper. In the proposed PSO, two representative positions, which represent the features of all personal-best positions, are defined to acquiring the information of all personal-best positions. Then the cognition term in velocity update equation is formed by three positions. Due to the effect of all personal-best fitnesses, each particle can update its velocity and position by the distribution of personal-best fitnesses. This strategy could make full use of all personal-best information and correct some error guided directions of personal-best positions.

The structure of rest paper is as follows. Section "Particle swarm optimizer" presents the theory and formulation of PSO algorithm and linearly decreasing inertia weight. In

section “Particle swarm optimization using all personal-best information”, the details of two representative positions are described and the proposed PSO using several multi-information characteristics of all personal-best positions is provided. Numerical results and statistical analysis are shown in section “Experiments and results”. In section “Conclusions”, we conclude this paper.

Particle swarm optimizer

Velocity and position formulation

Particle swarm optimizer is inspired by fish’s and birds’ foraging behaviors, which are simplified as a swarm of particles by mimicking their key behaviors. As a swarm of n particles search in the feasible space, each particle’s position represents a potential solution for the optimization problem and the swarm can find high-quality solution though the particles update their velocities and positions. Assuming the decision has m variables, the position and velocity of particle i are represented by m -dimensional vector $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{im})$ and $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{im})$. Two positions, named personal-best position and global-best position, are defined in PSO to update the velocities and guide the swarm. Personal-best position of particle i is denoted as $\mathbf{p}_{\text{best},i} = (p_{\text{best},i1}, p_{\text{best},i2}, \dots, p_{\text{best},im})$ and global-best position of particle i is denoted as $\mathbf{g}_{\text{best}} = (g_{\text{best},1}, g_{\text{best},2}, \dots, g_{\text{best},m})$. To sum up, the formulations of the velocity \mathbf{v}_i^{t+1} and the position \mathbf{x}_i^{t+1} of particle i can be expressed by the Eq. (1) and (2).

$$\mathbf{v}_i^{t+1} = \omega \mathbf{v}_i^t + c_1 r_1 (\mathbf{p}_{\text{best},i}^t - \mathbf{x}_i^t) + c_2 r_2 (\mathbf{g}_{\text{best}}^t - \mathbf{x}_i^t) \quad (1)$$

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \quad (2)$$

where c_1, c_2 are cognitive factor and social factor. ω is inertia weight. r_1, r_2 are two real numbers randomly in $(0, 1)$. t is the current generation. According to the theory of PSO, the personal experience and global experience make the particle move closer to them to get a new promising position.

Linearly decreasing inertia weight

Appropriate selection of inertia weight can balance global exploration and local exploitation during the evolution process. Large ω can benefit the global search while small value can contribute to local exploitation. Linearly decreasing inertia weight adopted in PSO (LPSO) significantly improves the performance of PSO for solving various optimization problems and the inertia weight ω is advised by the Eq. (3):

$$\omega = \omega_{\text{max}} - (\omega_{\text{max}} - \omega_{\text{min}}) \frac{t}{T} \quad (3)$$

where T is the maximal generation. ω_{min} and ω_{max} are the upper limit and lower limit. Numerical experiments illustrated the impact of ω , and 0.9 (upper value) and 0.4 (lower value) are suggested (Shi and Eberhart 1999).

From the above description, LPSO pseudo-code is shown in Algorithm 1.

Algorithm 1 LPSO pseudo-code

```

01: begin
02: Initialize particles' positions  $\mathbf{x}$  and velocities  $\mathbf{v}$ , find  $\mathbf{p}_{\text{best}}$  and  $\mathbf{g}_{\text{best}}$ .
03:  $t=1$ 
03: while ( $t \leq T$ )
04:    $\omega \leftarrow \omega_{\text{max}} - (\omega_{\text{max}} - \omega_{\text{min}}) * t / T$ .
04:   for  $i=1$  to  $n$ 
05:      $\mathbf{v}_i^{t+1} \leftarrow \omega \mathbf{v}_i^t + c_1 r_1 (\mathbf{p}_{\text{best},i}^t - \mathbf{x}_i^t) + c_2 r_2 (\mathbf{g}_{\text{best}}^t - \mathbf{x}_i^t)$ ;
06:      $\mathbf{x}_i^{t+1} \leftarrow \mathbf{x}_i^t + \mathbf{v}_i^{t+1}$ 
06:     Evaluate the fitness of  $\mathbf{x}_i^{t+1}$ , and update  $\mathbf{p}_{\text{best},i}$  and  $\mathbf{g}_{\text{best}}$ .
08:   end
09:    $t=t+1$ 
10: end
11: end

```

Particle swarm optimization using all personal-best information

Analysis of personal-best information

Learning behavior is a special skill for social animals, which can share the information with their members. Cooperative behavior of a swarm is more efficient than one taking an action alone due to their fruitful information and communication. In PSO, each particle can provide its personal-best position information to guide its flying direction. The whole personal-best positions of the swarm imply the distribution of fruitful good-fitness-related information. To take full advantage of multi-information characteristics of all personal-best information will contribute to ignoring several particles' error information trapping in local optima. In the theory of PSO, personal-best position is only used for its own particle in evolutionary process, not reflecting the influence of fitness distribution in landscape. Misguided information of personal-best positions, which have no opportunities to be corrected, will make PSO premature. Therefore, two positions, which add the influence of personal-best fitness distribution, are defined to strengthen the particle's ability to learn from other particles' experience. Then cognition term with three defined personal-best positions in velocity update equation is formed to reduce the misguided opportunity. The details of the improved cognition term and the proposed PSO algorithm are as follows.

Detail of improved PSO algorithm

Step 1 Calculate all personal-best positions' fitnesses, and then figure out the minimal fitness and the maximal fitness among these personal-best fitnesses. The way to find the minimal fitness and the maximal fitness is as follows:

$$f_{\min} = \min\{f(\mathbf{p}_{\text{best},i}) | i = 1, 2, \dots, n\} \tag{4}$$

$$f_{\max} = \max\{f(\mathbf{p}_{\text{best},i}) | i = 1, 2, \dots, n\} \tag{5}$$

where f_{\min} and f_{\max} stand for the minimal fitness and the maximal fitness of personal-best positions. f denotes the fitness function.

Step 2 Normalization method of personal-best fitness.

As the fitness value varies with a wide range in various optimization problems, a robust way to suitably reflect the influence of fitness is to normalize personal-best fitness. For minimization problem, the smaller the fitness value, the stronger the influence of personal-best position. According this feature, the normalization method is as Eq. (6).

$$r_i = \frac{f_{\max} - f(\mathbf{p}_{\text{best},i})}{f_{\max} - f_{\min}} \tag{6}$$

where r_i stands for the normalized value of the i th personal-best fitness.

Step 3 After normalized the personal-best fitness, we should also acquire the proportions of these fitnesses. The proportion of the i th personal-best fitness is denoted as θ_i . Thus, for the normalized value of the i th personal-best fitness, θ_i can be obtained as follows:

$$\theta_i = \begin{cases} r_i / \sum_{i=1}^n r_i & \text{if } f_{\max} \neq f_{\min} \\ 1/n & \text{otherwise} \end{cases}$$

Step 4 Calculate centroid position $\mathbf{p}_{\text{centr}}$ of all personal-best positions:

$$p_{\text{centr},j} = \sum_{i=1}^n p_{\text{best},ij} \theta_i \tag{7}$$

Centroid position is defined as weighted sum form of \mathbf{p}_{best} with θ to reflect the influence of personal-best fitnesses. Similar to the relation of an object’s density and mass in physics, by regarding personal-best fitness as ‘the density of an object’ and personal-best position as ‘the location in the object’, the position $\mathbf{p}_{\text{centr}}$ can be seen as ‘the centroid of the object’. The centroid of an object is important factor to reflect the distribution of mass and thus $\mathbf{p}_{\text{centr}}$ reflects the distribution of high quality fitness. The centroid position is always close to the area where most good fitnesses locate.

Step 5 Calculate median position \mathbf{p}_{med} of all personal-best positions.

\mathbf{p}_{med} represents the position of the median personal-best fitness. \mathbf{p}_{med} also reflects the distribution of high quality fitness from another perspective. \mathbf{p}_{med} is obtained without weighted sum form and can avoid the influence of some bad personal-best positions. Algorithm 2 is Pseudo-code to find the median fitness θ_{med} and the median position \mathbf{p}_{med} .

Algorithm 2 Pseudo-code of finding θ_{med} and p_{med} .

```

01: begin
02:   if mod(n,2) == 0                               % n is an even number.
03:     [order, index] = sort(theta, 'descend');      % Sort theta in descending order.
04:     theta_med = theta_n/2;
05:     p_med = p_best(index_n/2, :);
06:   else                                             % n is an odd number.
07:     theta_med = median(theta);                    % Find the median of personal-best fitnesses.
08:     index = find(theta == theta_med);            % Find the particle index of the median fitness
09:     p_med = p_best(index_1, :);                  % Obtain the position of the median fitness
10:   end
11: end

```

Step 6 Cognitive guiding position p'_{best} .

In the proposed PSO, cognitive guiding position using the above defined positions is calculated according to the following equations:

$$p'_{best,i} = \frac{p_{best,i} + p_{centr} - p_{med}}{2} \tag{8}$$

The cognitive guiding position includes three positions, the personal-best position p_{best} , the centroid position p_{centr} and the median position p_{med} . p_{best} and $p_{centr} - p_{med}$ are used to ‘pull’ the particle to escape local optimum because some error information of p_{best} and g_{best} may accelerate premature convergence. p_{centr} and p_{med} carry all personal-best information and can guide particles to a better direction. The experimental coefficient of 1/2 makes the cognitive guiding position suitable for the improved cognition term.

Step 7 Improved cognition term a_{cog} .

$$a_{cog,i} = \sum_{i=1}^n p'_{best,i} \theta_i - x_i \tag{9}$$

The improved cognition term a_{cog} will make full use of all personal-best fitnesses.

Step 8 Modified velocity update equation.

In this step, the work is to replace original cognition term with the improved cognition term a_{cog} in velocity update equation of PSO and LPSO algorithm. Therefore, particle swarm optimizer using multi-information characteristics of all personal-best information (PSO-API) and Linearly decreasing inertia weight PSO-API (LPSO-API) can be obtained using this modified velocity update equation. Take LPSO-API algorithm for example, each particle’s velocity updates as Eq. (10).

$$v_i^{t+1} = \omega v_i^t + r_1 \cdot a_{cog,i}^t + c \cdot r_2 \cdot (g_{best}^t - x_i^t) \tag{10}$$

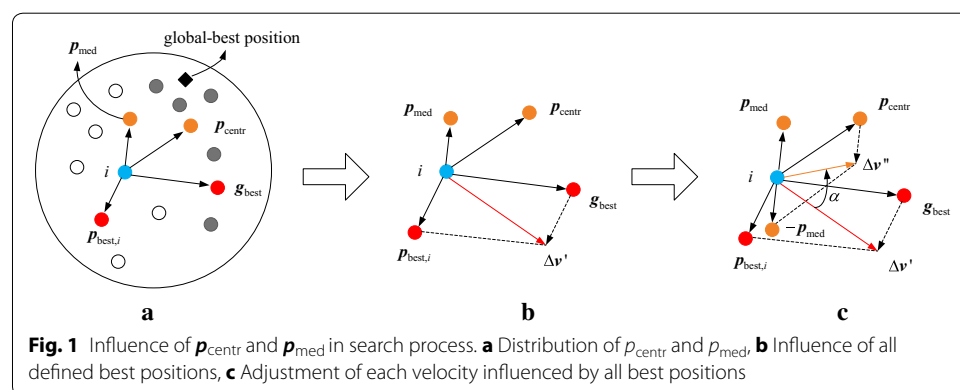
Not considering the influence of the current velocity and all the coefficients, there are four positions (p_{best} , g_{best} , p_{centr} and p_{med}) to influence the velocity update in Eq. (10).

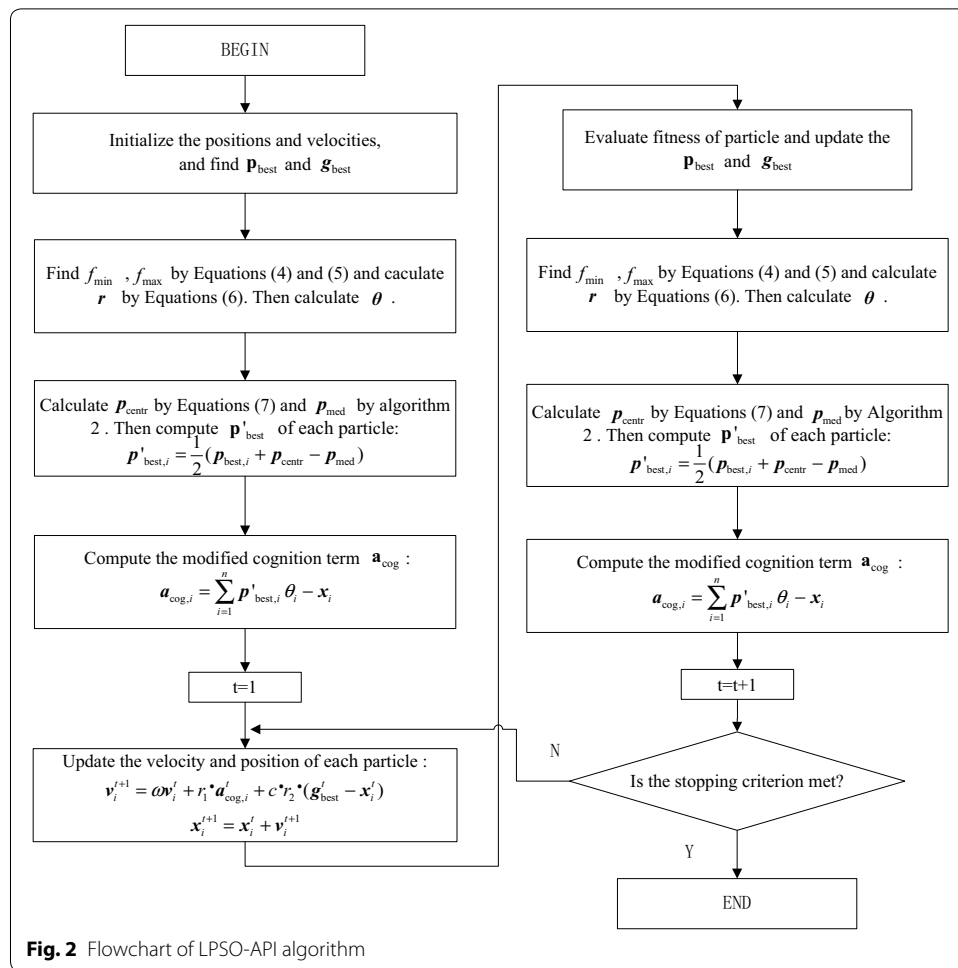
In PSO, $\Delta v' = g_{best} + p_{best,i}$ is introduced to show the influence of g_{best} and $p_{best,i}$. As is illustrated in Fig. 1b, if the current iteration g_{best} is local optimum, $\Delta v'$ will accelerate the particles fall into local optimum region. Comparing with PSO, p_{centr} and p_{med} is added to velocity update equation in PSO-API. In Fig. 1a, the white circle points represents the personal-best positions with worse fitnesses and the grey circle points represents the personal-best positions with better fitnesses. From the distribution of these above points, the location of p_{centr} and p_{med} , which are calculated by Eq. (7) and Algorithm 2, are shown by the yellow circle points in Fig. 1. Defined by all personal-best positions and their fitnesses, p_{centr} is more close to the region which locates many personal-best positions with good fitnesses. Although the fitness around the real global-best position is worse than that of local optimum g_{best} , the personal-best positions are also prone to distribute in these positions with good fitnesses around real global-best position, which is the black rhombic point in Fig. 1a. Regard p_{centr} as a reference point, and $\Delta v'' = p_{best,i} - p_{med}$, which carries all personal-best information, represents the influence of good fitness distribution. As is illustrated in Fig. 1c, α represents the direction adjusted by $\Delta v''$ and $\Delta v''$ makes particles adjust their directions to the real global-best position. Constantly adjusted by α in the search process, the particles have a greater probability of flying to the real global-best position. Besides, $|\Delta v''|$ will be small value when an uniform fitness distribution occurs in the search process and $|\Delta v''|$ makes little effect on particles. That is, PSO-API only has p_{best} and g_{best} influence particles' trajectory and PSO-API has the same performance as PSO in that case. Therefore, three terms ($\Delta v''$, p_{best} and g_{best}) contribute to adjusting the velocity and different 'pull' and 'push' influence make PSO-API have a stable performance over a variety of problems. The flowchart of LPSO-API algorithm is shown in Fig. 2.

Experiments and results

Test benchmark functions

In order to assess the performance of the proposed algorithm, twenty benchmark problems including unimodal, multimodal, rotated and shifted functions selected from the literature (Deep and Thakur 2007; Liang et al. 2006; Suganthan et al. 2005; Yao et al. 1999) are used to verify it. Note that all the problems are minimum problems and only one global optimum exists. The function name, dimensions, search range and global optimum value are listed in Table 1 and the formulations of these problems are listed as follows:





1. Sphere Function (unimodal function)

$$f_1(x) = \sum_{i=1}^n x_i^2$$

2. Schwefel's Problem 2.22 (unimodal function)

$$f_2(x) = \sum_{i=1}^n |x_i| + \prod_{i=1}^n |x_i|$$

3. Schwefel's Problem 1.2 (unimodal function)

$$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$$

4. Schwefel's Problem 2.21 (unimodal function)

$$f_4(x) = \max_i \{ |x_i|, 1 \leq i \leq n \}$$

Table 1 Twenty benchmark problems

Function name	Dimensions	Search range	Global optimum
$f_1(x)$	20/30/50	$[-100, 100]^D$	0
$f_2(x)$	20/30/50	$[-10, 10]^D$	0
$f_3(x)$	20/30/50	$[-100, 100]^D$	0
$f_4(x)$	20/30/50	$[-100, 100]^D$	0
$f_5(x)$	20/30/50	$[-100, 100]^D$	0
$f_6(x)$	20/30/50	$[-1.28, 1.28]^D$	0
$f_7(x)$	20/30/50	$[-5.12, 5.12]^D$	0
$f_8(x)$	20/30/50	$[-5.12, 5.12]^D$	0
$f_9(x)$	20/30/50	$[-32, 32]^D$	0
$f_{10}(x)$	20/30/50	$[-600, 600]^D$	0
$f_{11}(x)$	20/30/50	$[-0.5, 0.5]^D$	0
$f_{12}(x)$	20/30/50	$[-50, 50]^D$	0
$f_{13}(x)$	20/30/50	$[-1, 1]^D$	0
$f_{14}(x)$	20/30/50	$[-5.12, 5.12]^D$	0
$f_{15}(x)$	20/30/50	$[-100, 100]^D$	0
$f_{16}(x)$	20/30/50	$[-100, 100]^D$	0
$f_{17}(x)$	20/30/50	$[-1.28, 1.28]^D$	0
$f_{18}(x)$	20/30/50	$[-100, 100]^D$	-450
$f_{19}(x)$	20/30/50	$[-32, 32]^D$	-140
$f_{20}(x)$	20/30/50	$[-0.5, 0.5]^D$	90

5. Step Function (unimodal function)

$$f_5(x) = \sum_{i=1}^n (|x_i + 0.5|)^2$$

6. Quartic Function, i.e. Noise (unimodal function)

$$f_6(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1)$$

7. Generalized Rastrigin's Function (multimodal function)

$$f_7(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$$

8. Non-continuous Rastrigin's Function (multimodal function)

$$f_8(x) = \sum_{i=1}^n [y_i^2 - 10 \cos(2\pi y_i) + 10]$$

$$\text{where } y_i = \begin{cases} x_i & |x_i| \leq 0.5 \\ \frac{\text{round}(2x_i)}{2} & |x_i| \geq 0.5 \end{cases}$$

9. Ackley's Function (multimodal function)

$$f_9(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i \right) + 20 + e$$

10. Generalized Griewank Function (multimodal function)

$$f_{10}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

11. Weierstrass Function (multimodal function)

$$f_{11}(x) = \sum_{i=1}^n \left(\sum_{K=0}^{k_{\max}} \left[a^k \cos\left(2\pi b^k (x_i + 0.5)\right) \right] \right) - n \sum_{K=0}^{k_{\max}} \left[a^k \cos\left(2\pi b^k \times 0.5\right) \right]$$

where $a = 0.5, \quad b = 3, \quad k_{\max} = 20$

12. Generalized Penalized Function (multimodal function)

$$f_{12}(x) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 \left[1 + 10 \sin^2(\pi y_{i+1}) \right] + (y_n - 1)^2 \right\}$$

$$+ \sum_{i=1}^n u(x_i, 10, 100, 4) y_i = 1 + \frac{x_i + 1}{4}, u(x_i, a, k, m)$$

$$= \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < a \end{cases}$$

13. Cosine mixture Problem (multimodal function)

$$f_{13}(x) = \sum_{i=1}^n x_i^2 - 0.1 \sum_{i=1}^n \cos(5\pi x_i)$$

14. Rotated Rastrign Function (multimodal function)

$$f_{14}(x) = \sum_{i=1}^n \left[y_i^2 - 10 \cos(2\pi y_i) + 10 \right], \quad y = M \times x$$

15. Rotated Salomon Function (multimodal function)

$$f_{15}(x) = 1 - \cos\left(2\pi \sqrt{\sum_{i=1}^n y_i^2}\right) + 0.1 \sqrt{\sum_{i=1}^n y_i^2}, \quad y = M \times x$$

16. Rotated Rosenbrock Function (multimodal function)

$$f_{16}(x) = \sum_{i=1}^{n-1} \left[100 \left(y_i^2 - y_{i+1} \right)^2 + (y_i - 1)^2 \right], \quad y = M \times x$$

17. Rotated Elliptic Function (unimodal function)

$$f_{17}(x) = \sum_{i=1}^n \left(10^6 \right)^{(i-1)/(n-1)} y_i^2, \quad y = M \times x$$

18. Shifted Schewefel's Problem 2.21 (unimodal function)

$$f_{18}(x) = \max_i \{|y_i|, 1 \leq i \leq n\} + fbias_{18}, \quad y = x - o$$

where $fbias_{18} = -450$.

19. Shifted Rotated Ackley's Function (multimodal function)

$$f_{19}(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n z_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi z_i \right) + 20 + e + fbias_{19}$$

where $fbias_{19} = -140$, $z = (x - o) \times M'$

20. Shifted Rotated Weierstrass Function (multimodal function)

$$f_{20}(x) = \sum_{i=1}^n \left(\sum_{K=0}^{kmax} [a^k \cos(2\pi b^k(z_i + 0.5))] \right) - n \sum_{K=0}^{kmax} [a^k \cos(2\pi b^k \times 0.5)] + fbias_{20}$$

where $a = 0.5$, $b = 3$, $kmax = 20$, $fbias_{20} = 90$, $z = (x - o) \times M'$

Experimental analysis

Validity of the proposed strategy

To validate the proposed strategy, PSO-API and LPSO-API algorithms are implemented on matlab 2011a to compare with PSO and LPSO algorithms. All twenty benchmarks are tested in the experiments. Parameter settings of the four algorithms are as follows: The size of the population is 30. c_1 and c_2 are both equal to 2 in PSO and LPSO algorithms, and c is equal to 2 in PSO-API and LPSO-API algorithms. ω is equal to 0.7 in PSO and PSO-API algorithms and uses the suggested linearly decreasing version of section "Linearly decreasing inertia weight" in LPSO and LPSO-API algorithms (Shi and Eberhart 1999). 20, 30 and 50 dimensions are adopted in our experiments and the generations are 5000. Also, 20 independent trials are implemented on these problems. Tables 7, 9 and 11 in "Appendix" show the comparisons of 20, 30, 50 dimensions' results of average best fitness(AVE), rank(Rank) of average best fitness, median best fitness (MED), standard deviation (SD), average rank (AR) and final rank (FR) of average best fitness.

Wilcoxon's rank sum test is commonly used to analyze whether two data sets are statistically different from each other, and p value(p), h-value(h) and $zval(z)$ are acquired in Wilcoxon's rank sum test. In this test, significance level needs to be set and a value of 0.05 significance level indicates that something occurs more than the probability of 95 %. In Wilcoxon's rank sum test, h-value only has three value, 1, 0, -1, which indicate that the proposed algorithm have a significantly better, same and worse performance than the compared algorithm, respectively (Beheshti et al. 2013). Tables 8, 10 and 12 in "Appendix" show the comparisons of 20, 30, 50 dimensions' results of Wilcoxon's rank sum test. In details, the last three rows of Tables 8, 10 and 12 list the numbers of 1, 0 or -1 that h-value equals. Note that the best results for each benchmark function are marked in bold in Tables 7-12.

From 20, 30 and 50 dimensions' results in Tables 7, 9 and 11, it is clearly that LPSO-API algorithm obtains the minimum value in terms of AVE on twelve, fourteen, and fifteen of twenty benchmark problems, respectively and PSO-API algorithm obtains the minimum value of AVE on ten, ten and eight of twenty benchmark problems, respectively. It is obviously that LPSO-API algorithm and PSO-API algorithm obtains more

minimum results than LPSO algorithm and PSO algorithm in terms of AVE on the suite of benchmark problems. It is worth pointing out that several global optimums are also obtained by LPSO-API algorithm and PSO-API algorithm. The numbers of best AVE obtained by four algorithms are described in Fig. 3.

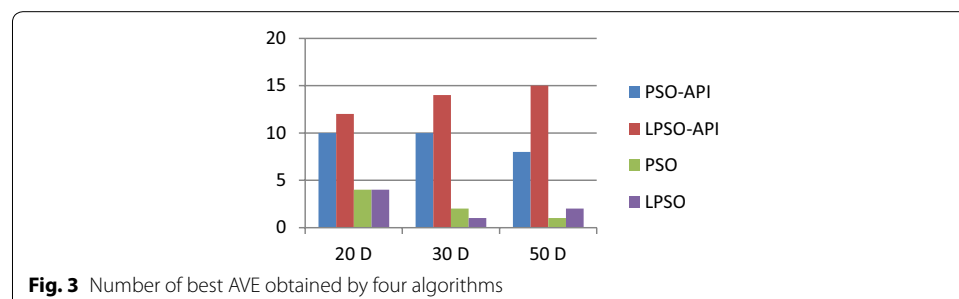
For three different dimensions, final rank obtained by LPSO-API algorithm all takes the first place and that obtained by PSO-API algorithm are all the second. The final rank can reflect the comprehensive performance of algorithm on a suite of benchmark problems. From the rank, it is clearly seen that LPSO-API algorithm and PSO-API algorithm shows the superiority than LPSO algorithm and PSO algorithm in high-quality solutions.

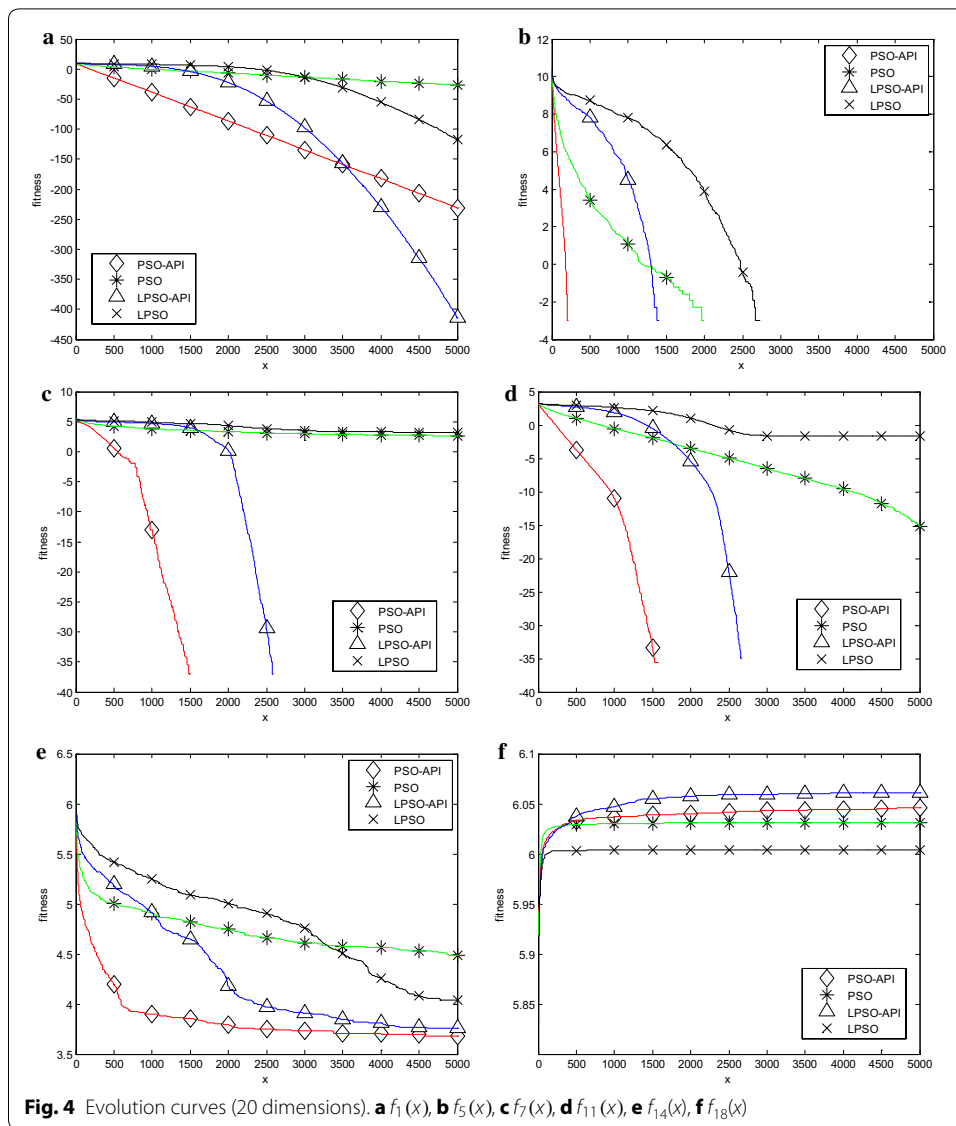
From the data in Tables 8, 10 and 12, the number of h-value = 1 is 16, 17 and 17 for PSO-API algorithm and 13, 14 and 17 for LPSO-API algorithm on 20, 30 and 50 dimensions' problems. A few h-value = -1 and h-value = 0 exist. It means that the results of LPSO-API and PSO-API algorithm statistically significantly outperform that of the PSO and LPSO algorithm. Also, by comparing the number of h-value = 1 with 20, 30, 50 dimensions' problems, we can see that the higher the dimension, the larger the number of h-value = 1 of LPSO-API algorithm and PSO-API algorithm. It illustrates that the LPSO-API algorithm and PSO-API algorithm perform better on high-dimension problem than low-dimension problem to some degree. From the above analysis, the proposed strategy of using all personal-best information is valid and efficient for solving most optimization problems, especially in high dimensions.

Six representative benchmark problems, two unimodal problems $f_1(x)$ and $f_5(x)$, two multimodal problems $f_7(x)$ and $f_{11}(x)$, a rotated problems $f_{14}(x)$, a shifted problems $f_{18}(x)$ are chosen for describing the process of fitness evolution. The evolutions of average fitness on these six problems are shown in Figs. 4a–f, 5a–f, 6a–f, respectively. Note that it is the logarithm of average fitness on vertical axis. It is clearly seen from these figures that PSO-API algorithm and LPSO-API algorithm obtain better solution with a fast convergence speed.

Comparison experiments with other PSO variants

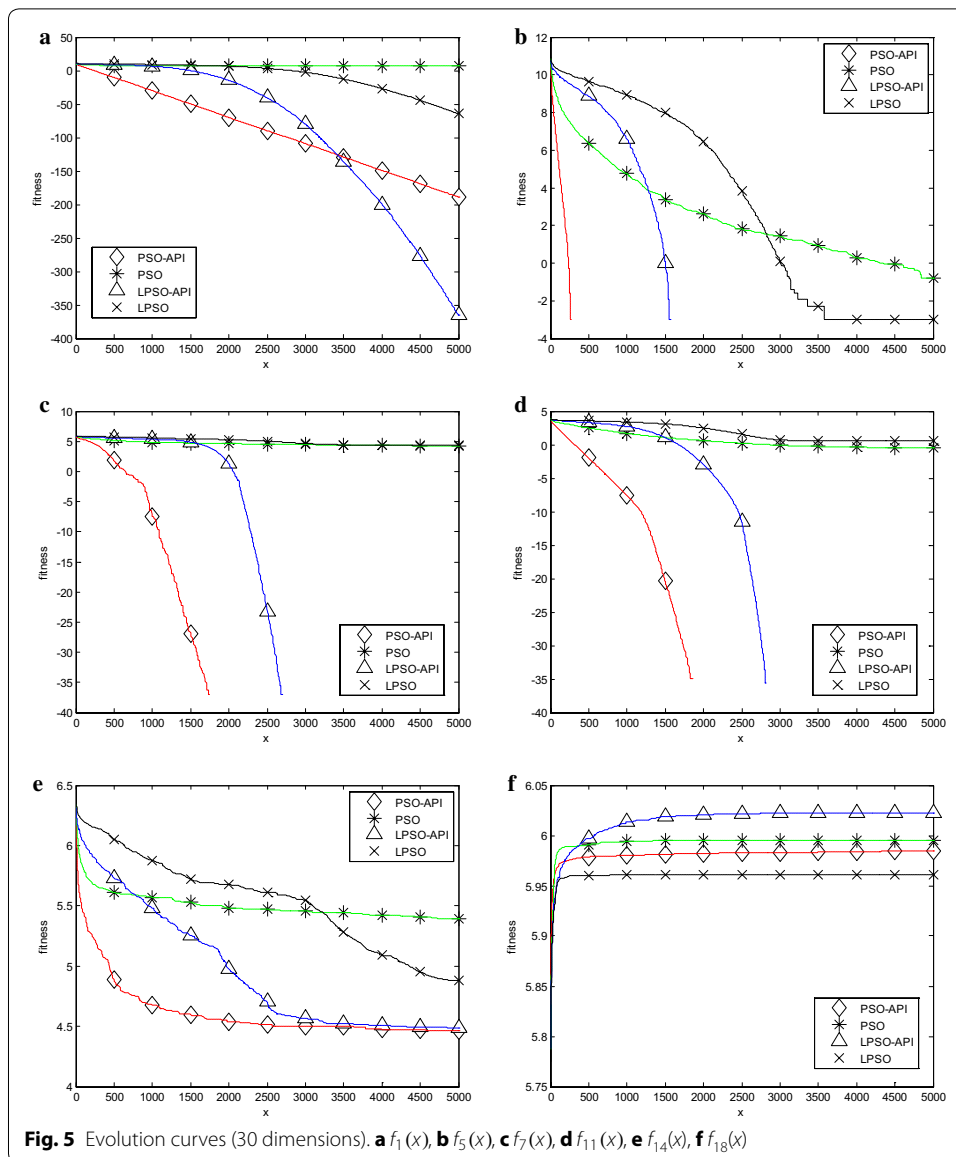
In recent literatures, various PSO algorithms are also developed and perform well on numerical experiments. To compare with these PSO algorithms, eight PSO variants (PSO-cf (Kennedy and Mendes 2002), FIPS (Mendes et al. 2004), HPSO-TVAC (Ratnaweera et al. 2004), VPSO (Kennedy and Mendes 2006), DMS-PSO (Liang and Suganthan 2005), CLPSO (Liang et al. 2006) and APSO (Zhan et al. 2009) are introduced to optimize ten benchmark functions, which are $f_1(x)$, $f_2(x)$, $f_3(x)$, $f_5(x)$, $f_6(x)$, $f_7(x)$, $f_8(x)$, $f_9(x)$, $f_{10}(x)$ and $f_{12}(x)$ in section “Test benchmark functions”. Table 2 shows their parameters settings and their results are from the corresponding paper (Zhan et al. 2009). The





generations are 2×10^5 and dimension number is 30. The size of population is 20. All the problems are optimized 30 times. Parameters settings of PSO-API and other settings are identical to that in last section. The comparisons of these PSO algorithms are shown in Table 3 in terms of average best fitness (Best) and standard deviation (SD), rank (Rank), average rank (AR) and final rank (FR) of average best fitness. Note that the best results for each benchmark function are marked in bold in Table 3.

From Table 3, the data of Rank demonstrates that PSO-API algorithm obtains best results on $f_1(x)$, $f_2(x)$, $f_3(x)$, $f_5(x)$, $f_7(x)$, $f_8(x)$, $f_9(x)$, $f_{10}(x)$ and performs worst on $f_6(x)$ and $f_{12}(x)$. Table 3 also shows FR obtained by PSO-API algorithms is better than that obtained by other eight PSO variants. It can be concluded that PSO-API algorithm has the highest comprehensive performance among them. Consequently, the comparisons indicate that PSO-API algorithm has the best overall performance over several existing PSO variants and is an effective method for solving a variety of optimization problems.



Time complexity of algorithm also should be considered and a computational experiment of six PSO variants [PSO-cf (Kennedy and Mendes 2002), FIPS (Mendes et al. 2004), DMS-PSO (Liang and Suganthan 2005), CLPSO (Liang et al. 2006), LPSO (Shi and Eberhart 1998c) and PSO-API] is performed over 20 independent runs and the execution times of these algorithms are compared. In the experiment, parameters settings of these algorithms are the same as Table 2. The population, dimension and generations are 20, 30 and 3000, respectively. Table 4 lists CPU times (in seconds) of six PSO algorithms. In Table 4, ‘AV(CPU)’ and ‘Rank’ stand for the average CPU time over 20 runs and the ascending order of each ‘AV(CPU)’, respectively. ‘AR’ and ‘FR’ stand for the average rank of Rank and the ascending order of AR, respectively.

From Rank of LPSO and PSO-API algorithm, we can conclude that our proposed policy adding to the original PSO increases the computational time. In Table 4, AR reflects comprehensive time-consuming order of the algorithm for twenty benchmarks. From

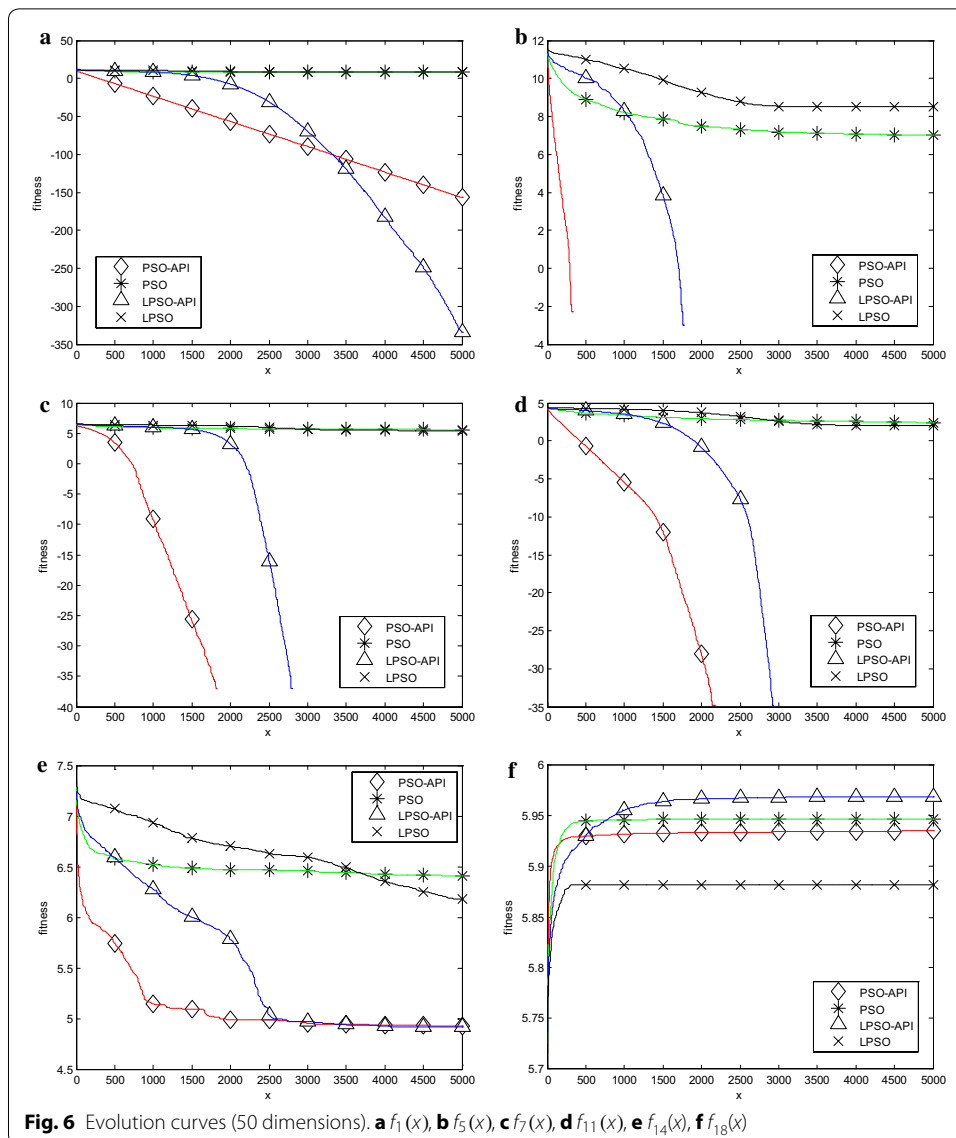


Table 4, the value of AR for PSO-cf and LPSO are smallest among all six algorithms. It illustrates that PSO-cf and LPSO, which are better than our proposed algorithm, have the best CPU time. The value of ‘AR’ for PSO-API algorithm and CLPSO are highly close to each other and it demonstrates that they have similar overall time consumption. The value of ‘AR’ for PSO-cf and LPSO are ‘4.7’ and ‘5.45’, which are both worse than PSO-API algorithm. From the value of ‘FR’, although PSO-API algorithm only ranks four, it is worthy of spending time to improve the accuracy of PSO algorithm. It is clear from the above comparisons of the accuracy and time consumption that PSO-API algorithm has a good overall balance between the performance and time complexity.

Comparisons experiments with similar PSO algorithms

In order to compare with FSS (Carmelo Filho et al. 2008) and CenterPSO (Liu et al. 2007), several experiments are carried out in this section. To compare with FSS algorithm, the

Table 2 Parameters settings of PSO variants

PSO variant	Topology	Parameters settings
PSO-cf	Local ring	$\omega: 0.9 - 0.4, c_1 = c_2 = 2.0$
FIPS	Local ring	$\chi = 0.729, \sum c_i = 4.1$
HPSO-TVAC	Global star	$\omega: 0.9 - 0.4, c_1: 2.5 - 0.5, c_2: 0.5 - 2.5$
DMS-PSO	Dynamic multi-swarm	$\omega: 0.9 - 0.4, m = 3, R = 5$
VPSO	Local von neumann	$\omega: 0.9 - 0.4, c_1 = c_2 = 2.0$
CLPSO	Comprehensive learning	$\omega: 0.9 - 0.4, C = 1.49455, m = 7$
APSO	Global star	$\omega : 0.9, c_1 = c_2 = 2.0, \delta : \text{random in } [0.050, 1], \sigma : 1 - 0.1$

Table 3 Numerical results for the comparisons

Name		PSO-cf	FIPS	HPSO-TVAC	DMS-PSO	VPSO	CLPSO	APSO	PSO-API
$f_1(x)$	Best	4.77e-29	3.21e-30	3.38e-41	3.85e-54	5.11e-38	1.89e-19	1.45e-150	0.00
	SD	1.13e-28	1.91e-30	8.50e-41	1.75e-53	1.91e-37	1.49e-19	5.73e-150	0.00
	Rank	7	6	4	3	5	8	2	1
$f_2(x)$	Best	2.03e-20	1.32e-17	6.9e-23	2.61e-29	6.29e-27	1.01e-13	5.15e-84	3.95e-323
	SD	2.89e-20	7.86e-18	6.89e-23	6.6e-29	8.68e-27	6.51e-14	1.44e-83	5.13e-322
	Rank	6	7	5	3	4	8	2	1
$f_3(x)$	Best	18.60	0.77	2.89e-7	47.5	1.44	395	1.0e-10	0.00
	SD	30.71	0.86	2.97e-7	56.4	1.55	142	2.13e-10	0.00
	Rank	6	4	3	7	5	8	2	1
$f_5(x)$	Best	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	SD	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Rank	1	1	1	1	1	1	1	1
$f_6(x)$	Best	1.49e-2	2.55e-3	5.54e-2	1.1e-2	1.08e-2	3.92e-3	4.66e-3	5.88e-001
	SD	5.66e-3	6.25e-4	2.08e-2	3.94e-3	3.24e-3	1.14e-3	1.7e-3	2.73e-001
	Rank	6	1	7	5	4	2	3	8
$f_7(x)$	Best	34.90	29.98	2.39	28.1	34.09	2.57e-11	5.8e-15	0.00
	SD	7.25	10.92	3.71	6.42	8.07	6.64e-11	1.01e-14	0.00
	Rank	8	6	4	5	7	3	2	1
$f_8(x)$	Best	30.40	21.33	35.91	1.83	32.8	0.167	4.14e-16	0.00
	SD	9.23	9.46	9.49	2.65	6.49	0.397	1.45e-15	0.00
	Rank	6	5	8	4	7	3	2	1
$f_9(x)$	Best	1.85e-14	7.69e-15	2.06e-10	8.52e-15	1.14e-14	2.01e-12	1.11e-14	3.55e-015
	SD	4.80e-15	9.33e-16	9.45e-10	1.79e-15	3.48e-15	9.22e-13	3.55e-15	0.00e+000
	Rank	6	2	8	3	5	7	4	1
$f_{10}(x)$	Best	1.10e-2	9.04e-4	1.07e-2	1.31e-2	1.31e-2	6.45e-13	1.67e-2	0.00
	SD	1.60e-2	2.78e-3	1.14e-2	1.73e-2	1.35e-2	2.07e-12	2.41e-2	0.00
	Rank	5	3	4	7	6	2	8	1
$f_{12}(x)$	Best	2.18e-30	1.22e-31	7.07e-30	2.05e-32	3.46e-3	1.59e-21	3.76e-31	9.72e-002
	SD	5.14e-30	4.85e-32	4.05e-30	8.12e-33	1.89e-2	1.93e-21	1.2e-30	1.97e-002
	Rank	4	2	5	1	7	6	3	8
	AR	5.5	3.7	4.9	3.9	5.1	4.8	2.9	2.4
	FR	8	3	6	4	7	5	2	1

experiments settings are as follows: five benchmarks with 30 dimensions are used to assess the algorithms. In detail, Generalized Rosenbrock Function and $f_3(x), f_7(x), f_9(x), f_{10}(x)$ in section “[Test benchmark functions](#)” are introduced. Generalized Rosenbrock Function is denoted as $f_{21}(x)$, and the expression of $f_{21}(x)$ is shown as follows.

Table 4 Computational time of six PSO algorithms

Function		PSO-cf	FIPS	DMS-PSO	CLPSO	LPSO	PSO-API
$f_1(x)$	AV(CPU)/Rank	6.09e-001/1	4.19e+000/6	4.04e+000/5	3.60e+000/4	7.59e-001/2	1.28e+000/3
$f_2(x)$	AV(CPU)/Rank	1.76e+000/3	4.10e+000/5	4.85e+000/6	3.55e+000/4	9.99e-001/1	1.51e+000/2
$f_3(x)$	AV(CPU)/Rank	1.01e+001/2	1.16e+001/3	1.28e+001/4	9.95e+000/1	1.44e+001/5	1.61e+001/6
$f_4(x)$	AV(CPU)/Rank	3.06e+000/3	4.19e+000/5	5.50e+000/6	3.67e+000/4	1.13e+000/1	1.66e+000/2
$f_5(x)$	AV(CPU)/Rank	3.31e+000/3	4.12e+000/6	4.11e+000/5	3.78e+000/4	8.22e-001/1	1.27e+000/2
$f_6(x)$	AV(CPU)/Rank	5.78e+000/1	7.04e+000/4	8.51e+000/6	6.79e+000/3	6.51e+000/2	7.35e+000/5
$f_7(x)$	AV(CPU)/Rank	3.17e+000/3	4.49e+000/5	4.56e+000/6	3.98e+000/4	1.03e+000/1	1.46e+000/2
$f_8(x)$	AV(CPU)/Rank	5.05e+000/2	6.59e+000/5	6.96e+000/6	5.89e+000/4	4.72e+000/1	5.31e+000/3
$f_9(x)$	AV(CPU)/Rank	4.83e+000/3	6.31e+000/5	7.23e+000/6	5.66e+000/4	3.18e+000/1	3.89e+000/2
$f_{10}(x)$	AV(CPU)/Rank	4.21e+000/1	6.53e+000/5	6.83e+000/6	6.16e+000/4	4.74e+000/2	5.20e+000/3
$f_{11}(x)$	AV(CPU)/Rank	5.09e+001/2	5.17e+001/3	6.52e+001/4	4.78e+001/1	9.52e+001/5	9.80e+001/6
$f_{12}(x)$	AV(CPU)/Rank	6.19e+000/1	1.22e+001/3	1.25e+001/4	1.12e+001/2	1.80e+001/6	1.75e+001/5
$f_{13}(x)$	AV(CPU)/Rank	4.52e-002/1	3.98e+000/6	3.91e+000/5	3.51e+000/4	1.07e+000/2	1.46e+000/3
$f_{14}(x)$	AV(CPU)/Rank	3.66e+000/3	4.78e+000/5	4.97e+000/6	4.24e+000/4	2.51e+000/1	2.94e+000/2
$f_{15}(x)$	AV(CPU)/Rank	3.81e+000/3	4.90e+000/5	5.00e+000/6	4.47e+000/4	2.72e+000/1	3.15e+000/2
$f_{16}(x)$	AV(CPU)/Rank	4.49e+000/2	5.57e+000/5	6.62e+000/6	5.10e+000/4	4.14e+000/1	4.51e+000/3
$f_{17}(x)$	AV(CPU)/Rank	4.80e+000/3	5.90e+000/5	1.02e+001/6	4.67e+000/1	4.79e+000/2	5.35e+000/4
$f_{18}(x)$	AV(CPU)/Rank	3.12e-003/1	4.60e+000/5	5.70e+000/6	3.67e+000/4	2.19e+000/2	2.58e+000/3
$f_{19}(x)$	AV(CPU)/Rank	1.56e-003/1	5.84e+000/5	6.25e+000/6	4.35e+000/3	4.13e+000/2	4.57e+000/4
$f_{20}(x)$	AV(CPU)/Rank	2.65e+001/2	2.76e+001/3	3.25e+001/4	2.00e+001/1	4.99e+001/6	4.95e+001/5
	AR	2.05	4.7	5.45	3.2	2.25	3.35
	FR	1	5	6	3	2	4

$$f_{21}(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \quad (-100 \leq x_i \leq 100)$$

Population size of PSO-API sets as 30. 30 runs are conducted for each problem and each run will perform 1×10^4 generations. To compare with CenterPSO algorithm, the experiments settings are as follows: Three benchmarks $f_7(x), f_{10}(x), f_{21}(x)$ with 30 dimensions are used. The generations is 2000. The population has four sizes of 20, 40, 80, 160. Each experiment will perform 100 runs. Average best fitness (Avg. best fitness) and standard deviation (SD) of PSO-API, FSS and CenterPSO are presented in Tables 5 and 6. It's worth noting that the better results are marked in bold in Tables 5 and 6.

From the data in Table 5, it can be seen that PSO-API obtains better average best fitness and standard deviation than that obtained by FSS algorithm for all five benchmarks except Generalized Rosenbrock Function. However, for Generalized Rosenbrock Function, PSO-API and FSS algorithm obtain the results with the same order of magnitude. From the Table 6, the results obtained by PSO-API with all population sizes are better than that obtained by CenterPSO algorithm for all three benchmarks. Therefore, statistics analysis indicates the proposed algorithm have better performance than FSS algorithm and CenterPSO algorithm. For most of the benchmarks, all above experiments indicates that PSO-API is a high-performance algorithm.

Table 5 Comparison results with FSS algorithm

Name		PSO-API	FSS
$f_3(x)$	Avg. best fitness/SD	3.883e-090/9.563e-090	8.080e-002/2.200e-002
$f_7(x)$	Avg. best fitness/SD	0.000e+000/0.000e+000	1.338e+001/4.005e+000
$f_9(x)$	Avg. best fitness/SD	3.789e-015/9.013e-016	4.000e-002/2.000e-002
$f_{10}(x)$	Avg. best fitness/SD	0.000e+000/0.000e+000	2.700e-003/2.000e-003
$f_{21}(x)$	Avg. best fitness/SD	2.635e+001/ 3.145e-001	1.611e+001/7.290e-001

Table 6 Comparison results with CenterPSO algorithm

Name	Size		PSO-API	CenterPSO
$f_7(x)$	20	Avg. best fitness/SD	0.000e+000/0.000e+000	3.359e+001/9.562e+000
	40	Avg. best fitness/SD	0.000e+000/0.000e+000	2.668e+001/7.764e+000
	80	Avg. best fitness/SD	0.000e+000/0.000e+000	2.276e+001/6.758e+000
	160	Avg. best fitness/SD	2.020e-010/2.020e-009	2.141e+001/5.949e+000
$f_{10}(x)$	20	Avg. best fitness/SD	2.311e-004/1.711e-003	1.200e-002/1.650e-002
	40	Avg. best fitness/SD	7.841e-005/7.841e-004	8.800e-003/1.190e-002
	80	Avg. best fitness/SD	8.442e-006/8.442e-005	9.300e-003/1.200e-002
	160	Avg. best fitness/SD	7.308e-015/7.151e-014	1.200e-002/1.680e-002
$f_{21}(x)$	20	Avg. best fitness/SD	2.702e+001/3.831e-001	1.319e+002/1.358e+002
	40	Avg. best fitness/SD	2.649e+001/3.276e-001	8.717e+001/6.365e+001
	80	Avg. best fitness/SD	2.626e+001/1.987e-001	6.234e+001/5.940e+001
	160	Avg. best fitness/SD	2.601e+001/2.379e-001	4.299e+001/4.499e+001

Conclusions

In this work, to make full use of multi-information characteristics of all personal-best information, an improved PSO algorithm using three positions with all personal-best information has been adopted to enhance the performance. In proposed algorithm, an improved cognition term using the personal-best position, the centroid position and the median position is introduced in velocity update process of PSO. To validate this strategy, a set of benchmark functions including unimodal, multimodal, rotated and shifted benchmark functions with 20, 30 and 50 dimensions have been optimized. Experimental results show that the strategy using multi-information characteristics of all personal-best information is a valid strategy for the purposes of improving the PSO’s performance. Moreover, PSO-API algorithm has also been used to compare with several PSO variants and some similar algorithms of the proposed algorithm. Numerical results show that the PSO-API algorithm has higher precision and satisfied performance. To sum up, the proposed strategy enhances the search ability of PSO and PSO-API algorithm is an efficient PSO variant to obtain promising solution for most of benchmark functions.

Authors’ contributions

S.H. carried out the study, collected data, designed the experiments, implemented the simulation, analyzed data and wrote the main manuscript. N.T. provided some intellectual information and revised the manuscript. Y.W. gave technical support and helped to the design of the study. Z.J. made the general supervision of the research. All authors read and approved the final manuscript.

Author details

¹ School of Internet of Things Engineering, Jiangnan University, 1800 Lihu Avenue, Wuxi 214122, Jiangsu Province, China. ² Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China. ³ Department of Educational Technology, Jiangnan University, 1800 Lihu Avenue, Wuxi, Jiangsu Province 214122, China.

Competing interests

The authors declare that they have no competing interests in this study.

Funding

In this work, the design of the study and collection, analysis, and interpretation of data are funded by the National High-tech Research and Development Projects of China under Grant No: 2014AA041505 and the writing of the manuscript is funded by the National Natural Science Foundation of China under Grant No: 61572238 and by the Provincial Outstanding Youth Foundation of Jiangsu Province under Grant No: BK20160001.

Appendix

See Tables 7, 8, 9, 10, 11, 12.

Table 7 Results of twenty benchmark problems (generations = 5000 and dimensions = 20)

Name		PSO-API	PSO	LPSO-API	LPSO
$f_1(x)$	AVE/rank	6.143e-101/2	3.714e-012/4	1.992e-180/1	5.879e-052/3
	MED/SD	2.803e-102/1.498e-100	6.501e-013/8.397e-012	4.447e-183/2.124e-180	1.476e-053/1.638e-051
$f_2(x)$	AVE/rank	1.522e-050/2	1.000e+000/3	3.973e-091/1	3.000e+000/4
	MED/SD	7.388e-051/2.773e-050	1.387e-010/3.077e+000	7.565e-092/1.174e-090	2.895e-033/4.701e+000
$f_3(x)$	AVE/rank	1.049e-060/2	3.145e+003/3	3.938e-132/1	8.000e+003/4
	MED/SD	8.599e-062/3.394e-060	2.604e+003/3.448e+003	5.529e-134/1.488e-131	1.000e+004/5.342e+003
$f_4(x)$	AVE/rank	8.837e-042/2	2.666e+000/4	1.290e-078/1	5.100e-004/3
	MED/SD	1.414e-042/1.569e-041	2.376e+000/1.136e+000	1.071e-079/4.198e-078	3.224e-004/6.771e-004
$f_5(x)$	AVE/rank	0.0e+000/1	0.0e+000/1	0.0e+000/1	0.0e+000/1
	MED/SD	0.0e+000/0.0e+000	0.0e+000/0.0e+000	0.0e+000/0.0e+000	0.0e+000/0.0e+000
$f_6(x)$	AVE/rank	4.992e-001/3	7.748e-001/4	4.983e-001/2	4.625e-001/1
	MED/SD	5.276e-001/3.032e-001	8.512e-001/3.869e-001	5.067e-001/2.680e-001	5.071e-001/2.670e-001
$f_7(x)$	AVE/rank	0.0e+000/1	1.411e+001/3	0.0e+000/1	2.490e+001/4
	MED/SD	0.0e+000/0.0e+000	9.455e+001/1.125e+001	0.0e+000/0.0e+000	1.542e+011/1.996e+001
$f_8(x)$	AVE/rank	0.0e+000/1	2.034e+001/3	1.773e+000/2	2.530e+001/4
	MED/SD	0.0e+000/0.0e+000	1.867e+011/1.158e+001	0.0e+000/6.846e+000	2.700e+011/1.999e+001
$f_9(x)$	AVE/rank	3.552e-015/1	2.440e-007/3	3.552e-015/1	7.676e-001/4
	MED/SD	3.552e-015/0.0e+000	1.615e-007/2.245e-007	3.552e-015/0.0e+000	7.105e-015/3.432e+000
$f_{10}(x)$	AVE/rank	0.0e+000/1	4.861e-002/4	7.404e-004/2	3.588e-002/3
	MED/SD	0.0e+000/0.0e+000	4.785e-002/3.737e-002	0.0e+000/3.311e-003	2.834e-002/3.710e-002
$f_{11}(x)$	AVE/rank	0.0e+000/1	2.615e-007/3	0.0e+000/1	2.000e-001/4
	MED/SD	0.0e+000/0.0e+000	1.650e-008/9.516e-007	0.0e+000/0.0e+000	0.000e+000/8.944e-001
$f_{12}(x)$	AVE/rank	5.335e-002/4	3.893e-008/1	5.103e-002/3	7.775e-003/2
	MED/SD	5.383e-002/9.641e-003	6.931e-014/1.741e-007	4.906e-002/1.051e-002	2.355e-032/3.477e-002
$f_{13}(x)$	AVE/rank	-2.0e+000/1	-2.0e+000/1	-2.0e+000/1	-2.0e+000/1
	MED/SD	-2.0e+000/0.0e+000	-2.0e+000/3.520e-015	-2.0e+000/0.0e+000	-2.0e+000/0.0e+000
$f_{14}(x)$	AVE/rank	3.963e+001/1	8.864e+001/4	4.278e+001/2	5.657e+001/3
	MED/SD	3.702e+011/1.884e+001	8.617e+011/3.462e+001	4.612e+011/1.646e+001	5.323e+011/2.357e+001
$f_{15}(x)$	AVE/rank	9.987e-002/1	1.370e+000/4	9.987e-002/1	1.199e+000/3
	MED/SD	9.987e-002/2.883e-010	4.998e-001/3.682e+000	9.987e-002/4.206e-010	3.998e-001/3.579e+000
$f_{16}(x)$	AVE/rank	1.876e+001/1	3.382e+007/3	1.877e+001/2	4.691e+008/4
	MED/SD	1.876e+001/2.135e-002	1.952e+031/1.374e+008	1.877e+001/1.374e-002	1.971e+08/5.627e+008
$f_{17}(x)$	AVE/rank	1.529e-012/2	1.252e+004/4	8.245e-035/1	8.891e+003/3
	MED/SD	1.549e-032/6.838e-012	4.387e+031/1.629e+004	8.006e-096/3.687e-034	6.856e+031/8.742e+003
$f_{18}(x)$	AVE/rank	-4.224e+002/2	-4.161e+002/3	-4.288e+002/1	-4.051e+002/4
	MED/SD	-4.229e+02/3.728e+000	-4.173e+02/1.819e+001	-4.290e+02/2.251e+000	-4.077e+02/1.290e+001
$f_{19}(x)$	AVE/rank	-1.1921e+2/4	-1.1931e+2/1	-1.1925e+2/3	-1.1928e+2/2
	MED/SD	-1.1920e+2/6.117e-002	-1.1928e+2/1.049e-001	-1.1923e+2/5.970e-002	-1.1928e+2/7.102e-002
$f_{20}(x)$	AVE/rank	1.0953e+002/4	1.047e+002/2	1.0951e+002/3	1.042e+002/1
	MED/SD	1.095e+02/2.107e+000	1.050e+02/2.457e+000	1.094e+002/2.029e+000	1.037e+02/2.714e+000
	AR	1.85	2.9	1.55	2.9
	FR	2	3	1	3

Table 8 Wilcoxon’s rank sum test results (generations = 5000 and dimensions = 20)

Name		PSO-API and PSO	LPSO-API and LPSO
$f_1(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.795e-008/1/-5.396e+000
$f_2(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.690e-008/1/-5.399e+000
$f_3(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.795e-008/1/-5.396e+000
$f_4(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.795e-008/1/-5.396e+000
$f_5(x)$	p/h/z	0.000e+000/0/0.000e+000	0.000e+000/0/0.000e+000
$f_6(x)$	p/h/z	3.151e-002/1/-2.150e+000	7.352e-001/0/3.381e-001
$f_7(x)$	p/h/z	8.006e-009/1/-5.768e+000	8.006e-009/1/-5.768e+000
$f_8(x)$	p/h/z	8.006e-009/1/-5.768e+000	3.519e-007/1/-5.093e+000
$f_9(x)$	p/h/z	8.006e-009/1/-5.768e+000	2.412e-008/1/-5.579e+000
$f_{10}(x)$	p/h/z	8.006e-009/1/-5.768e+000	2.859e-007/1/-5.132e+000
$f_{11}(x)$	p/h/z	8.006e-009/1/-5.768e+000	1.625e-001/1/-1.396e+000
$f_{12}(x)$	p/h/z	6.795e-008/1/5.396e+000	2.776e-007/1/5.137e+000
$f_{13}(x)$	p/h/z	3.122e-006/1/-4.662e+000	0.000e+000/0/0.000e+000
$f_{14}(x)$	p/h/z	3.293e-005/1/-4.152e+000	9.090e-002/0/-1.690e+000
$f_{15}(x)$	p/h/z	6.709e-008/1/-5.398e+000	5.612e-008/1/-5.430e+000
$f_{16}(x)$	p/h/z	1.200e-006/1/-4.855e+000	6.795e-008/1/-5.396e+000
$f_{17}(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.795e-008/1/-5.396e+000
$f_{18}(x)$	p/h/z	1.402e-001/0/-1.474e+000	1.025e-007/1/-5.322e+000
$f_{19}(x)$	p/h/z	3.381e-004/-1/3.584e+000	1.555e-001/0/1.420e+000
$f_{20}(x)$	p/h/z	3.499e-006/-1/4.639e+000	2.061e-006/-1/4.747e+000
	1	16	13
	0	2	6
	-1	2	1

Table 9 Results of twenty benchmark problems (generations = 5000 and dimensions = 30)

Name	PSO-API	PSO	LPSO-API	LPSO
$f_1(x)$	AVE/rank 1.499e-082/2	1.500e+003/4	2.291e-159/1	1.184e-028/3
	MED/SD 9.419e-084/3.947e-082	1.782e-003/3.663e+003	6.706e-163/9.471e-159	2.128e-029/3.017e-028
$f_2(x)$	AVE/rank 4.736e-042/2	5.000e+000/3	5.600e-082/1	1.450e+001/4
	MED/SD 2.354e-042/6.150e-042	4.860e-004/6.069e+000	2.822e-082/6.334e-082	1.500e+001/9.986e+000
$f_3(x)$	AVE/rank 1.395e-043/2	1.510e+004/3	2.721e-108/1	1.660e+004/4
	MED/SD 1.715e-044/2.946e-043	1.580e+004/6.208e+003	1.049e-109/5.623e-108	1.666e+004/5.629e+003
$f_4(x)$	AVE/rank 5.252e-032/2	1.769e+001/4	1.338e-066/1	1.968e+000/3
	MED/SD 1.688e-032/8.290e-032	1.719e+001/3.485e+000	8.649e-068/3.874e-066	1.846e+000/7.950e-001
$f_5(x)$	AVE/rank 0.000e+000/1	4.500e-001/4	0.000e+000/1	5.000e-002/3
	MED/SD 0.000e+000/0.000e+000	0.000e+000/8.255e-001	0.000e+000/0.000e+000	0.000e+000/2.236e-001
$f_6(x)$	AVE/rank 5.199e-001/2	2.045e+000/3	4.889e-001/1	2.307e+000/4
	MED/SD 5.157e-001/ 3.248e-001	1.456e+000/1.958e+000	4.368e-001/3.293e-001	6.323e-001/4.144e+000
$f_7(x)$	AVE/rank 0.000e+000/1	6.980e+001/3	0.000e+000/1	7.686e+001/4
	MED/SD 0.000e+000/0.000e+000	7.259e+001/2.296e+001	0.000e+000/0.000e+000	7.522e+001/2.848e+001
$f_8(x)$	AVE/rank 0.000e+000/1	1.018e+002/4	0.000e+000/1	8.270e+001/3
	MED/SD 0.000e+000/0.000e+000	9.411e+001/3.081e+001	0.000e+000/0.000e+000	8.001e+001/3.588e+001
$f_9(x)$	AVE/rank 3.552e-015/1	7.264e-001/3	3.552e-015/1	3.807e+000/4
	MED/SD 3.552e-015/0.000e+000	7.494e-003/3.189e+000	3.552e-015/0.000e+000	2.131e-014/6.874e+000
$f_{10}(x)$	AVE/rank 0.000e+000/1	5.139e-002/3	3.720e-004/2	1.354e+001/4
	MED/SD 0.000e+000/0.000e+000	4.340e-002/5.443e-002	0.000e+000/1.664e-003	1.845e-002/3.301e+001
$f_{11}(x)$	AVE/rank 0.000e+000/1	6.248e-001/3	0.000e+000/1	1.805e+000/4
	MED/SD 0.000e+000/0.000e+000	2.362e-002/1.952e+000	0.000e+000/0.000e+000	2.395e-003/3.033e+000
$f_{12}(x)$	AVE/rank 1.078e-001/3	2.256e-001/4	1.067e-001/2	5.183e-003/1
	MED/SD 1.040e-001/ 1.480e-002	1.098e-002/5.944e-001	1.030e-001/1.504e-002	1.634e-028/2.318e-002
$f_{13}(x)$	AVE/rank -3.000e+000/1	-2.999e+000/3	-3.000e+000/1	-2.940e+000/4
	MED/SD -3.000e+00/0.000e+000	-2.999e+000/7.685e-006	-3.000e+00/0.000e+000	-3.000e+00/2.683e-001
$f_{14}(x)$	AVE/rank 8.703e+001/1	2.200e+002/4	8.865e+001/2	1.319e+002/3
	MED/SD 9.867e+001/4.002e+001	2.177e+002/ 3.608e+001	1.101e+002/4.941e+001	1.186e+002/6.466e+001
$f_{15}(x)$	AVE/rank 9.987e-002/1	2.446e+000/3	9.987e-002/1	1.025e+001/4
	MED/SD 9.987e-002/5.355e-010	2.350e+000/6.174e-001	9.987e-002/7.443e-010	1.529e+001/9.398e+000
$f_{16}(x)$	AVE/rank 2.869e+001/1	1.573e+009/3	2.870e+001/2	2.542e+009/4
	MED/SD 2.869e+001/1.449e-002	1.095e+006/3.957e+009	2.870e+001/1.788e-002	1.082e+009/4.418e+009
$f_{17}(x)$	AVE/rank 1.786e-036/2	4.287e+003/3	3.634e-104/1	9.561e+003/4
	MED/SD 1.651e-037/3.264e-036	3.560e+003/2.774e+003	2.468e-110/1.625e-103	7.780e+003/6.709e+003
$f_{18}(x)$	AVE/rank -3.972e+002/3	-4.016e+002/2	-4.127e+002/1	-3.880e+002/4
	MED/SD -3.965e+02/ 3.643e+000	-4.060e+02/1.933e+001	-4.127e+02/4.270e+000	-3.966e+02/2.133e+001
$f_{19}(x)$	AVE/rank -1.190e+002/4	-1.190e+002/1	-1.190e+002/3	-1.190e+002/2
	MED/SD -1.190e+002/5.490e-002	-1.190e+02/6.539e-002	-1.190e+002/4.999e-002	-1.190e+002/1.005e-001
$f_{20}(x)$	AVE/rank 1.244e+002/3	1.153e+002/1	1.270e+002/4	1.172e+002/2
	MED/SD 1.239e+002/2.857e+000	1.154e+002/3.703e+000	1.266e+002/ 2.562e+000	1.166e+002/2.931e+000
AR	1.75	3.05	1.45	3.4
FR	2	3	1	4

Table 10 Wilcoxon’s rank sum test results (generations = 5000 and dimensions = 30)

Name		PSO-API and PSO	LPSO-API and LPSO
$f_1(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.795e-008/1/-5.396e+000
$f_2(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.690e-008/1/-5.399e+000
$f_3(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.795e-008/1/-5.396e+000
$f_4(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.795e-008/1/-5.396e+000
$f_5(x)$	p/h/z	9.496e-003/1/-2.593e+000	3.421e-001/0/-9.500e-001
$f_6(x)$	p/h/z	1.575e-006/1/-4.801e+000	3.851e-002/1/-2.069e+000
$f_7(x)$	p/h/z	8.006e-009/1/-5.768e+000	8.006e-009/1/-5.768e+000
$f_8(x)$	p/h/z	8.006e-009/1/-5.768e+000	8.006e-009/1/-5.768e+000
$f_9(x)$	p/h/z	8.006e-009/1/-5.768e+000	6.764e-009/1/-5.796e+000
$f_{10}(x)$	p/h/z	8.006e-009/1/-5.768e+000	2.208e-006/1/-4.733e+000
$f_{11}(x)$	p/h/z	8.006e-009/1/-5.768e+000	3.494e-007/1/-5.094e+000
$f_{12}(x)$	p/h/z	7.711e-003/-1/2.664e+000	3.415e-007/-1/5.098e+000
$f_{13}(x)$	p/h/z	8.006e-009/1/-5.768e+000	1.625e-001/0/-1.396e+000
$f_{14}(x)$	p/h/z	6.757e-008/1/-5.397e+000	1.476e-001/0/-1.447e+000
$f_{15}(x)$	p/h/z	6.776e-008/1/-5.396e+000	6.766e-008/1/-5.397e+000
$f_{16}(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.795e-008/1/-5.396e+000
$f_{17}(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.795e-008/1/-5.396e+000
$f_{18}(x)$	p/h/z	1.545e-002/-1/2.421e+000	8.505e-006/1/-4.452e+000
$f_{19}(x)$	p/h/z	9.277e-005/1/3.908e+000	6.389e-002/0/1.852e+000`
$f_{20}(x)$	p/h/z	3.938e-007/-1/5.071e+000	9.172e-008/-1/5.342e+000
	1	17	14
	0	0	4
	-1	3	2

Table 11 Results of twenty benchmark problems (generations = 5000 and dimensions = 50)

Name	PSO-API	PSO	LPSO-API	LPSO	
$f_1(x)$	AVE/rank	8.111e-069/2	4.571e+003/4	6.638e-146/1	3.000e+003/3
	MED/SD	2.816e-069/1.213e-068	1.378e+002/6.841e+003	4.585e-152/2.968e-145	9.040e-012/4.701e+003
$f_2(x)$	AVE/rank	1.339e-035/2	2.635e+001/3	1.770e-076/1	3.800e+001/4
	MED/SD	6.996e-036/1.559e-035	3.020e+001/1.696e+001	5.086e-078/6.311e-076	3.000e+001/1.735e+001
$f_3(x)$	AVE/rank	7.320e-029/2	6.095e+004/3	6.921e-091/1	7.035e+004/4
	MED/SD	4.714e-031/3.121e-028	6.084e+004/1.020e+004	2.327e-092/1.678e-090	6.774e+004/2.032e+004
$f_4(x)$	AVE/rank	9.832e-024/2	4.042e+001/4	1.467e-055/1	2.482e+001/3
	MED/SD	3.494e-024/1.388e-023	4.021e+001/4.275e+000	1.603e-057/5.564e-055	2.621e+001/3.161e+000
$f_5(x)$	AVE/rank	0.000e+000/1	1.106e+003/3	0.000e+000/1	5.000e+003/4
	MED/SD	0.000e+000/0.000e+000	1.130e+002/3.090e+003	0.000e+000/0.000e+000	1.000e+000/6.069e+003
$f_6(x)$	AVE/rank	4.829e-001/1	1.498e+001/3	5.144e-001/2	1.795e+001/4
	MED/SD	4.569e-001/2.745e-001	1.618e+001/9.801e+000	5.114e-001/ 2.600e-001	1.068e+001/1.779e+001
$f_7(x)$	AVE/rank	0.000e+000/1	2.677e+002/4	0.000e+000/1	2.329e+002/3
	MED/SD	0.000e+000/0.000e+000	2.671e+002/5.346e+001	0.000e+000/0.000e+000	2.276e+002/4.966e+001
$f_8(x)$	AVE/rank	1.259e+001/2	2.729e+002/4	0.000e+000/1	2.725e+002/3
	MED/SD	0.000e+000/5.634e+001	2.803e+002/5.489e+001	0.000e+000/0.000e+000	2.680e+002/5.470e+001
$f_9(x)$	AVE/rank	4.973e-015/2	9.682e+000/3	4.618e-015/1	1.185e+001/4
	MED/SD	3.552e-015/1.785e-015	1.316e+001/5.389e+000	3.552e-015/1.670e-015	1.370e+001/7.483e+000
$f_{10}(x)$	AVE/rank	0.000e+000/1	2.885e+001/3	0.000e+000/1	3.159e+001/4
	MED/SD	0.000e+000/0.000e+000	2.355e+000/4.211e+001	0.000e+000/0.000e+000	1.599e-002/5.296e+001
$f_{11}(x)$	AVE/rank	0.000e+000/1	1.107e+001/4	0.000e+000/1	7.587e+000/3
	MED/SD	0.000e+000/0.000e+000	1.186e+001/3.728e+000	0.000e+000/0.000e+000	6.002e+000/4.384e+000
$f_{12}(x)$	AVE/rank	2.453e-001/3	5.749e+003/4	2.361e-001/2	9.655e-002/1
	MED/SD	2.475e-001/3.344e-002	1.219e+002/1.672e+004	2.427e-001/ 2.283e-002	6.220e-002/1.673e-001
$f_{13}(x)$	AVE/rank	-5.000e+000/1	-4.573e+000/4	-5.000e+000/1	-4.611e+000/3
	MED/SD	-5.000e+000/0.000e+000	-4.771e+000/4.670e-001	-5.000e+000/0.000e+000	-4.926e+000/8.788e-001
$f_{14}(x)$	AVE/rank	1.381e+002/2	6.056e+002/4	1.368e+002/1	4.808e+002/3
	MED/SD	2.115e+002/1.298e+002	5.709e+002/1.049e+002	2.102e+002/1.290e+002	4.337e+002/2.018e+002
$f_{15}(x)$	AVE/rank	9.987e-002/1	1.897e+001/3	9.987e-002/1	2.485e+001/4
	MED/SD	9.987e-002/5.210e-010	1.899e+001/6.756e+000	9.987e-002/5.908e-010	2.559e+001/7.030e+000
$f_{16}(x)$	AVE/rank	4.862e+001/1	1.186e+010/3	4.865e+001/2	1.631e+010/4
	MED/SD	4.862e+001/1.551e-002	4.668e+009/1.816e+010	4.864e+001/2.043e-002	1.336e+010/1.745e+010
$f_{17}(x)$	AVE/rank	6.371e-031/2	8.266e+004/3	7.198e-097/1	1.111e+005/4
	MED/SD	1.426e-031/1.347e-030	7.732e+004/4.103e+004	4.644e-104/3.219e-096	8.578e+004/8.496e+004
$f_{18}(x)$	AVE/rank	-3.782e+002/3	-3.824e+002/2	-3.911e+002/1	-3.586e+002/4
	MED/SD	-3.788e+002/ 1.934e+000	-3.834e+002/1.983e+001	-3.909e+002/3.716e+000	-3.539e+002/1.941e+001
$f_{19}(x)$	AVE/rank	-1.1883e+002/3	-1.1890e+002/1	-1.1882e+002/4	-1.1889e+002/2
	MED/SD	-1.1883e+02/ 3.887e-002	-1.1889e+02/3.944e-002	-1.1881e+02/4.171e-002	-1.1887e+02/7.086e-002
$f_{20}(x)$	AVE/rank	1.565e+002/3	1.435e+002/2	1.597e+002/4	1.426e+002/1
	MED/SD	1.566e+002/ 3.440e+000	1.421e+002/5.957e+000	1.590e+002/3.475e+000	1.436e+002/3.741e+000
AR		1.8	3.2	1.4	3.25
FR		2	3	1	4

Table 12 Wilcoxon’s rank sum test results (generations = 5000 and dimensions = 50)

Name		PSO-API and PSO	LPSO-API and LPSO
$f_1(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.786e-008/1/-5.396e+000
$f_2(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.795e-008/1/-5.396e+000
$f_3(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.795e-008/1/-5.396e+000
$f_4(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.795e-008/1/-5.396e+000
$f_5(x)$	p/h/z	8.006e-009/1/-5.768e+000	2.457e-005/1/-4.218e+000
$f_6(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.795e-008/1/-5.396e+000
$f_7(x)$	p/h/z	8.006e-009/1/-5.768e+000	8.006e-009/1/-5.768e+000
$f_8(x)$	p/h/z	3.046e-008/1/-5.538e+000	8.006e-009/1/-5.768e+000
$f_9(x)$	p/h/z	3.959e-008/1/-5.492e+000	3.290e-008/1/-5.525e+000
$f_{10}(x)$	p/h/z	8.006e-009/1/-5.768e+000	8.006e-009/1/-5.768e+000
$f_{11}(x)$	p/h/z	8.006e-009/1/-5.768e+000	8.006e-009/1/-5.768e+000
$f_{12}(x)$	p/h/z	6.795e-008/1/-5.396e+000	4.600e-004/-1/3.502e+000
$f_{13}(x)$	p/h/z	8.006e-009/1/-5.768e+000	7.991e-009/1/-5.768e+000
$f_{14}(x)$	p/h/z	5.726e-008/1/-5.427e+000	4.505e-007/1/-5.046e+000
$f_{15}(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.786e-008/1/-5.396e+000
$f_{16}(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.795e-008/1/-5.396e+000
$f_{17}(x)$	p/h/z	6.795e-008/1/-5.396e+000	6.795e-008/1/-5.396e+000
$f_{18}(x)$	p/h/z	1.055e-002/-1/2.557e+000	1.194e-006/1/-4.856e+000
$f_{19}(x)$	p/h/z	4.165e-005/-1/4.098e+000	4.600e-004/-1/3.502e+000
$f_{20}(x)$	p/h/z	3.938e-007/-1/5.071e+000	6.795e-008/1/-5.396e+000
	1	17	17
	0	0	0
	-1	3	3

Received: 9 May 2016 Accepted: 6 September 2016

Published online: 21 September 2016

References

Beheshti Z, Shamsuddin SMH, Hasan S (2013) MPSO: median-oriented particle swarm optimization. *Appl Math Comput* 219(11):5817–5836

Beielstein T, Parsopoulos KE, Vrahatis MN (2002) Tuning PSO parameters through sensitivity analysis. Universität Dortmund

Bonyadi MR, Li X, Michalewicz Z (2014) A hybrid particle swarm with a time-adaptive topology for constrained optimization. *Swarm Evol Comput* 18:22–37

Carmelo Filho JA, De Lima Neto FB, Lins AJCC et al (2008) A novel search algorithm based on fish school behavior. In: Proceedings of the 2008 IEEE international conference on systems, man and cybernetics, pp 2646–2651

Cheng R, Jin Y (2015) A social learning particle swarm optimization algorithm for scalable optimization. *Inf Sci* 291:43–60

Deep K, Thakur M (2007) A new crossover operator for real coded genetic algorithms. *Appl Math Comput* 188(1):895–911

Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, vol 1, pp 39–43

Haklı H, Uğuz H (2014) A novel particle swarm optimization algorithm with Levy flight. *Appl Soft Comput* 23:333–345

Hu W, Yen GG (2015) Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system. *IEEE Trans Evol Comput* 19(1):1–18

Kennedy ER (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, Perth, Australia, Piscataway, vol 4, 1942–1948

Kennedy J, Mendes R (2002) Population structure and particle swarm performance. In: Proceedings of the 2002 congress on evolutionary computation, vol 2, pp 1671–1676

Kennedy J, Mendes R (2006) Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *IEEE Trans Syst Man Cybern C Appl Rev* 36(4):515–519

Li Y, Zhan ZH, Lin S et al (2015) Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems. *Inf Sci* 293:370–382

- Liang JJ, Suganthan PN (2005) Dynamic multi-swarm particle swarm optimizer. In: Proceedings of the 2005 congress on swarm intelligence symposium, vol 8237, pp 124–129
- Liang JJ, Qin AK, Suganthan PN et al (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans Evol Comput* 10(3):281–295
- Lim WH, Isa NAM (2014a) An adaptive two-layer particle swarm optimization with elitist learning strategy. *Inf Sci* 273:49–72
- Lim WH, Isa NAM (2014b) Particle swarm optimization with adaptive time-varying topology connectivity. *Appl Soft Comput* 24:623–642
- Lim WH, Isa NAM (2014c) Particle swarm optimization with increasing topology connectivity. *Eng Appl Artif Intell* 27:80–102
- Lim WH, Isa NAM (2014d) Bidirectional teaching and peer-learning particle swarm optimization. *Inf Sci* 280:111–134
- Lim WH, Isa NAM (2014e) Teaching and peer-learning particle swarm optimization. *Appl Soft Comput* 18:39–58
- Liu Y, Qin Z, Shi Z et al (2007) Center particle swarm optimization. *Neurocomputing* 70(4):672–679
- Mendes R, Kennedy J, Neves J (2004) The fully informed particle swarm: simpler, maybe better. *IEEE Trans Evol Comput* 8(3):204–210
- Qin Q, Cheng S, Zhang Q et al (2015) Multiple strategies based orthogonal design particle swarm optimizer for numerical optimization. *Comput Oper Res* 60:91–110
- Rao RV, Patel V (2013) An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems. *Scientia Iranica* 20(3):710–720
- Ratnaweera A, Halgamuge SK, Watson HC (2004) Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans Evol Comput* 8(3):240–255
- Shi Y, Eberhart RC (1998) Parameter selection in particle swarm optimization. *Evolutionary programming VII*, vol 1447. Springer, Berlin, pp 591–600
- Shi Y, Eberhart R (1998b) A modified particle swarm optimizer. In: Proceedings of the 1998 IEEE international conference on evolutionary computation, vol 6, pp 69–73
- Shi Y, Eberhart R (1998c) A modified particle swarm optimizer. In: IEEE international conference on evolutionary computation, the 1998 IEEE international conference on computational intelligence, pp 69–73
- Shi Y, Eberhart RC (1999) Empirical study of particle swarm optimization. *Proc IEEE Congr Evol Comput* 3:1945–1950
- Suganthan PN, Hansen N, Liang JJ et al (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. In: Proceedings of IEEE congress on evolutionary computation, pp 1–50
- Sun S, Li J (2014) A two-swarm cooperative particle swarms optimization. *Swarm Evol Comput* 15:1–18
- Wang L, Yang B, Chen Y (2014) Improving particle swarm optimization using multi-layer searching strategy. *Inf Sci* 274:70–94
- Yadav A, Deep K (2014) An efficient co-swarm particle swarm optimization for non-linear constrained optimization. *J Comput Sci* 5(2):258–268
- Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. *IEEE Trans Evol Comput* 3(2):82–102
- Zhan ZH, Zhang J, Li Y et al (2009) Adaptive particle swarm optimization. *IEEE Trans Syst Man Cybern B Cybern* 39(6):1362–1381
- Zhang W, Ma D, Wei J et al (2014) A parameter selection strategy for particle swarm optimization based on particle positions. *Exp Syst Appl* 41(7):3576–3584

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
