Applied Soft Computing Journal 76 (2019) 45–52

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

Particle swarm optimization with damping factor and cooperative mechanism

Mingfu He [a], Mingzhe Liu [a,*], Ruili Wang [b], Xin Jiang [a], Bingqi Liu [a], Helen Zhou [c]

[a] State Key Laboratory of Geohazard Prevention and Geoenvironment Protection, Chengdu University of Technology, Chengdu 610059, China
[b] Institute of Natural and Mathematical Sciences, Massey University, Auckland, New Zealand
[c] School of Natural and Computational Sciences, Manukau Institute of Technology, Auckland, New Zealand

HIGHLIGHTS

- A new parameter, damping factor, is introduced to adjust the position information inherited from the previous state.
- The cooperative mechanism is employed to help find the global optima quickly.
- *Pleast*, is defined to decide whether current information of particles is abandoned and reinitialized.
- 24 benchmark functions and three variants of PSO are used to verify our approach.

ARTICLE INFO

Article history:
Received 16 April 2017
Received in revised form 4 October 2018
Accepted 30 November 2018
Available online 6 December 2018

Keywords:
Particle swarm optimization
Cooperative mechanism
Damping factor
C-means clustering

ABSTRACT

A novel variant of particle swarm optimization with damping factor and cooperation mechanism (PSO-DFCM) to search the global optima in a large scale and high-dimensional searching space. In this optimal searching strategy, one balances the exploring and exploiting abilities of particles by introducing a new parameter, a damping factor α, which is used to adjust the position information inherited from the previous state. The cooperative mechanism between the global-best-oriented and the local-best-oriented swarms is employed to help find the global optima quickly. In order to reduce the negative effect of unfavorable particles on swarm evolution, a new concept of evolution history, the least optimal particle in individuals' histories — *pleast*, is defined to decide whether current information of particles is abandoned and reinitialized in our proposal. Also, fuzzy c-means clustering is applied to cluster the particles' positions for the neighborhood establishment of individuals. Our comparative study on benchmark test functions demonstrates that the proposed PSO outperforms the standard PSO and three state-of-art variants of PSO in terms of global optimum convergence and final optimal results.

1. Introduction

Particle swarm optimization (PSO) was firstly proposed by Eberhart and Kennedy [1], inspired by biosocial phenomena such bird flocking and fish schooling. POS is a population-based stochastic optimization technique, the basic notion of which is that social sharing of information among peers provides a great evolutionary advantage. In PSO and its extended algorithms, swarm populations are the candidate solution space of *fitness* functions and the agents of swarm are correspondingly called particles. Animals, especially birds and fishes, normally travel in groups without colliding. Each member adjusts its position and velocity using the collective and historical information, hence it reduces the individual's effort for searching food or shelters. When compared with other evolutionary optimization algorithms, PSO has better computational efficiency for continuous optimal problems mainly because of less memory space requiring fewer model parameters to be adjusted and less effort to implement [2]. Thus, PSO and its extensions have many successful applications in practical engineering optimizations [3].

Most studies in the literature have focused on the parameter optimization for the standard PSO including the population size of swarm (*pop_size*), inertia weight (w), accelerating factors (c_1, c_2), constraint factor χ, velocity and position of particles (X_{max}, V_{max}) since the solution performance is sensitive to the selection of those parameters [4] (See Eq. (1)),

$$V_{i,j}^{t+1} = wV_{i,j}^{t} + c_1r_1(pbest(t)_{i,j} - X_{i,j}^{t}) + c_2r_2(gbest(t) - X_{i,j}^{t}) \quad (1a)$$

$$X_{i,j}^{t+1} = X_{i,j}^{t} + \chi V_{i,j}^{t+1} \quad (1b)$$

where $X_{i,j}^{t}$ ($|X_{i,j}^{t}| \leq X_{max}$) and $V_{i,j}^{t}$($|V_{i,j}^{t}| \leq V_{max}$) are the jth particle's velocity and position in the ith dimension at the time index t, respectively; $pbest(t)_{i,j}$ is the historical optimal position of particle j, and $gbest(t)$ is the collective optimal position.

* Corresponding author.
E-mail address: liumz@cdut.edu.cn (M. Liu).

https://doi.org/10.1016/j.asoc.2018.11.050
1568-4946/

Although many efforts have been put into developing new techniques for parameter selection for different optimal problems, most current techniques have not obviously improved the limitation of the standard PSO [5]. To improve the performance and convergence of PSO, many researchers have successfully developed new variants of PSOs based on the theories in other fields. These variants are mainly in the following three categories:

(1) biologically extended PSOs: PSO with an aging leader and challengers [6], social learning PSO [7], PSO with a cooperative approach [8], clubs-based PSO [9], group decision PSO [10], principle component PSO [11] and multigrouped PSO [12];
(2) physically extended PSOs: PSO with fine tuning operator [13], opposition-based PSO [14], PSO with Bayesian techniques [15], PSO with recombination and dynamic linkage discovery [16], PSO with fuzzy clustering [17], two-parts-divided PSO [18], chaos enhanced PSO [19] and gravitational global PSO [20];
(3) hybrid PSOs with other heuristic algorithms: PSO with simulated annealing and swarm core evolutionary [21], PSO with differential evolution [22], genetic learning PSO [23], hybrid PSO with artificial bee colony [24], hybrid PSO with artificial fish swarm [25].

Note that more computational effort is required when PSO is combined and/or hybridized with other theories and algorithms. In addition, velocity updating techniques are improved to avoid the premature convergence to local optimal points [26,27]. Starting from a time-varying discrete dynamical system and stochastic process, three papers have proved that PSO needs for convergence to optimal points which are based on *s* second order difference equation [28,29]. Tian [30] gives a review of convergence analysis in PSO and its extended algorithms.

This paper is motivated by the two issues of PSO and its variants: (i) difficulty in obtaining optima in a large-scale and high-dimensional space, and (ii) premature convergence to local optimal points [31]. A new damping factor α is used to balance the exploring and exploiting abilities of particles, and a cooperative mechanism between the global-best-oriented and the local-best-oriented swarms is employed to help find global optima more quickly. A parameter, the least optimal particles in individuals' histories, is adopted to decide whether current information of particles is abandoned and reinitialized in order to reduce the negative effect of unfavorable particles on swarm evolution. Also, fuzzy *c*-means clustering is applied to cluster the particles' positions for the individuals' neighborhood establishment in order to speed up convergence. The proposed approach PSO-DFCM has shown better performance in global optimum convergence and final optimal results, compared with the standard PSO and three state-of-art PSO versions.

The rest of this paper is organized as follows: Section 2 presents the algorithm by introducing the damping factor, cooperative mechanism and parameters choice. Section 3 discusses the experimental results. Concluding remarks are described in Section 4.

2. Methodology

2.1. Damping movement inspired by general inertia law

In reality, many equilibrating systems will generate some resistant mechanisms to transient external forces that break the previous equilibrium states and bring themselves back into a new equilibrium, which is called *General Inertia Law* [32]. It has alternative explanations in various fields: Newton's law in kinetics, Hooke's law in a spring system, Lenz's law in electromagnetics, Le Chatelier's principle in chemical reaction systems and Estrous Cycle in biology. In PSO, optimal solutions depend on the particles' positions, which are updated by particles' previous positions and current velocities. Thereby, one applies the inertia rule to refine the position update of each particle according to comparison between previous and current position of each particle, and Eq. (1b) can be transformed as follows:

$$X_{i,j}^{t+1} = \lambda^{\Delta_j^*} X_{i,j}^t + \chi V_{i,j}^{t+1} \tag{2}$$

where $\alpha = \lambda^{\Delta_j^*}$ $(\lambda > 1.0)$ is a damping factor, which decides how much inherited information particles enlarge or shrink from the previous states, and Δ_j $(\Delta_j = fitness_j^t - fitness_j^{t-1})$ is an inertia-balancing factor to decide whether individuals enlarge or diminish inherited information. Eq. (2) is the modification equation for particle position update from Eq. (1b), which is based on the logical analysis of inertia. It gives a description of how much position information passing from the previous state to the current state. In order to avoid the premature convergence of particles, Δ_j is normalized into $[-1, 1]$ as follows:

$$\Delta_j^* = \frac{2}{max\{\Delta_j\} - min\{\Delta_j\}}(\Delta_j - min\Delta_j) - 1 \tag{3}$$

and $max\{\Delta_j\} \neq min\{\Delta_j\}$, otherwise $\Delta_j^* = 0$. If $\Delta_j > 0$, the particle *j* searches the optimal points in an exploring mode, otherwise in an exploiting mode. Eq. (3) just defines inertia-balancing factor to decide whether individuals enlarge or diminish inherited information based on the previous state.

The exploring or exploiting mode is achieved by Eq. (3) according to the definition of the damping factor, when Δ_j is greater than zero, then particle *j* inherits more position information from its previous state and moves bigger steps, which means the particle *j* explores new surroundings. However when Δ_j is less than zero, the particle *j* just walks around the adjacent neighbors due to smaller movement, thus the particle *j* is in the mode of exploitation.

2.2. Cooperative mechanism between individuals

In a natural environment, birds often confront numerous challenges, such as food scarcity and low breeding rates. They normally cooperate with other members in neighborhoods to increase the foraging efficiency [33] and the chance of breed success [34]. Previous research has introduced some cooperative mechanisms for PSO improvement. However, these cooperative mechanisms are just limited to best individuals [35–37], rather than all individuals. This paper applied a modified cooperative mechanism to all individuals. The cooperative mechanism may have the following advantages: (1) all individuals can exchange the location information of the best particle by local and global search, this improvement can absorb mutual advantages of both local and global search modes; (2) through the competition strategy, the top *N* particles will be selected out of a group of 2*N* particles, therefore, the convergence to the global optima will be guaranteed.

In the standard PSO, $gbest(t)$ is the collective optimal point at time index *t*. Based on the neighborhood size, there are two versions: global version $gbest(t) \rightarrow gbest_g(t)$, and local version $gbest(t) \rightarrow gbest_l(t)$. $gbest_g(t)$ refers to the best particle candidate of the whole swarm while $gbest_l(t)$ represents the best particle of the individual's neighborhood. For swarm neighborhood calculation, the social network adopted by $gbest_g(t)$ reflects the star topology, which offers a faster convergence but it is very likely to converge prematurely. While $gbest_l(t)$ uses a local social network topology, where smaller neighborhoods are defined for each particle (e.g., ring, sphere, hexagon, cylinder). Due to the lower particle inter-connectivity of $gbest_l(t)$, it is less susceptible to being trapped in local minima at the cost of slow convergence. Herein, one aggravates two versions of $gbest(t)$ to take advantage of their mutual merits and inhibit their unfavorable factors. It is suggested that a dual swarm can be employed to search the global optima:

(1) global-best aimed swarm evolves in Eq. (4)

$$\begin{aligned} {}^gV_{i,j}^{t+1} &= \chi \times ({}^gV_{i,j}^t + c_1^g r_1^g (pbest(t)_{i,j} - X_{i,j}^t) \\ &\quad + c_2^g r_2^g (gbest_g(t) - X_{i,j}^t)) \end{aligned} \tag{4}$$

where $\chi = 2/|2 - c - \sqrt{c^2 - 4c}|(c = c_1^g + c_2^g)$;

(2) local-best aimed swarm evolves in Eq. (5)

$$\begin{aligned} {}^lV_{i,j}^{t+1} &= w(t) \times {}^lV_{i,j}^t + c_1^l r_1^l (pbest(t)_{i,j} - X_{i,j}^t) \\ &\quad + c_2^l r_2^l (gbest_l(t) - X_{i,j}^t) \end{aligned} \tag{5}$$

Eqs. (4) and (5) are the velocity equations of PSO for the global and local search mode, respectively. Note that fuzzy c-means clustering is utilized for the neighborhood establishment of $gbest_l(t)$. There may be a chance that $gbest_l(t)$s sometimes prematurely converge. Therefore, these $gbest_l(t)$s should escape from local optimal points by way of random walks. A perturbing term of random walk is incorporated into Eq. (5) and the new updating method of ${}^lV_{i,j}^{t+1}$ is as in Eq. (6)

$$\begin{aligned} {}^lV_{i,j}^{t+1} &= w(t) \times {}^lV_{i,j}^t + c_1^l r_1^l (pbest(t)_{i,j} - X_{i,j}^t) \\ &\quad + c_2^l r_2^l (gbest_l(t) - X_{i,j}^t) + R \times N(\mu, \Sigma) \end{aligned} \tag{6}$$

where $\mu = E[X]$, $\Sigma = COV(X)$ and R is a trigger factor of random walk which is defined by Eq. (7)

$$R = \begin{cases} 1 & r_1^l + r_2^l > r_1^g + r_2^g \\ 0 & otherwise \end{cases} \tag{7}$$

If $r_1^l + r_2^l > r_1^g + r_2^g$, the particles' learning abilities are overwhelmed and their obsessive tendencies are alleviated by random walk to avoid premature convergence. Eq. (6) is the modified equation of Eq. (5) incorporating the random walk for avoiding the local optima trap. The purpose of this equation is that using random walk allows the opportunity of local optima points. Eq. (7) is defined as a trigger factor for the random walk.

A cooperative mechanism among all individual particles is presented based on cooperative strategies as follows:

(1) compute particles' temporary positions individually according to Eqs. (4) and (5) as follows:

$${}^gXtemp_{i,j}^{t+1} = X_{i,j}^t \lambda^{\Delta_j} + {}^gV_{i,j}^{t+1} \tag{8a}$$

$${}^lXtemp_{i,j}^{t+1} = X_{i,j}^t \lambda^{\Delta_j} + {}^lV_{i,j}^{t+1} \tag{8b}$$

(2) update particles' positions based on overall competition of Eq. (9)

$$X_{i,j}^{t+1} = Xtemp_{i,j}^{t+1} \tag{9}$$

where $\forall Xtemp_{i,j}^{t+1}$ satisfies

$$\begin{aligned} & f(Xtemp_{:,1}^{t+1}) \le f(Xtemp_{:,2}^{t+1}) \le \cdots, f(Xtemp_{:,pop\_size}^{t+1}) \\ & \quad \le f(Xtemp_{:,pop\_size+1}^{t+1}) \le \cdots \le f(Xtemp_{:,2pop\_size}^{t+1}) \end{aligned}$$

(3) update particles' reference information: $pbest(t+1)_{i,j}$, $gbest_l(t+1)$ and $gbest_g(t+1)$ from $X_{i,j}^{t+1}$.

Eq. (8) is to compute the updated positions of both local and global search modes for providing the best candidates for the particle group. Eq. (9) describes how the best candidates are selected for the particle group.

2.3. Weak particles reinitialization tactic

In order to reduce the negative effect of weak particles, one also needs a weak-point-abandoned tactic to guarantee the swarm movement towards potential optimal direction as much as possible. Therefore, the least optimal particle in individual histories, $pleast_{:,j}(t)$, is defined, which updates as Eq. (10)

$$pleast_{:,j}(t+1) = \begin{cases} pleast_{:,j}(t) & f(pleast_{:,j}) \ge max\{f(X_{:,j})\} \\ X_{:,jmax} & otherwise \end{cases} \tag{10}$$

where $X_{:,jmax}$ satisfies $f(pleast_{:,jmax}) = max\{f(X_{:,j})\}$. Also, its position is to be re-initialized ($X_{:,j}^t \sim N(\mu, \Sigma)$) and if the particle's fitness value is less optimal than that of $pleast_{:,j}(t)$. Eq. (10) gives the location information of least optimal particle. The logic behind this weak particle tactic is to allow the particle to avoid the re-exploration in the unwanted zone that has been tracked before.

2.4. Parameters choice and algorithm initialization

In the method presented, $r_{1,2}^{l,g}$ are uniform random numbers ($\sim U(0, 1)$). It is simply assume that $pop\_size = 2N \times D + 20$; the number of clustering centers is N and the variable D is used to denote the dimension of the search space. For adaptive calculation of $w(t)$, one develops a diverse information entropy based inertia weight in virtue of *fuzzy c-means* clustering:

$$w(t) = \frac{1}{1 + a \times exp(-b \sum_{n=1}^{N} q_n \log(\frac{1}{q_n}))} \tag{11}$$

where q_n is the ratio of the number of clusters n to N; a and b are the constant numbers related to the weight range [0.4, 0.9]. Eq. (11) justifies the weight for the local search mode which is based on the definition of information entropy, and converts the information entropy to the weight range using a nonlinear function. In this paper, the negative exponential function is adopted due to the decay property. Other functions can also be considered. In Eq. (4), the particles explore coarsely the optima in global searching space, therefore, acceleration factors (c_1^g and c_2^g) should be more than 2.0 and remain constant. In our proposal, the default values for both c_1^g and c_2^g are set to be 2.05. In Eq. (5), the particles exploit the optima in a local foraging space in a delicate way, hence, acceleration factors (c_1^l and c_2^l) are respectively different over evolution and are supposed to be less than 2.0, the calculation of c_1^l and c_1^2 is as follows [38]:

$$c_1^l = 1.5 - \frac{min\{f_j(t)\}}{f_j(t)} \tag{12a}$$

$$c_2^l = 0.5 + \frac{min\{f_j(t)\}}{f_j(t)} \tag{12b}$$

Procedure of the proposed PSO-DFCM:

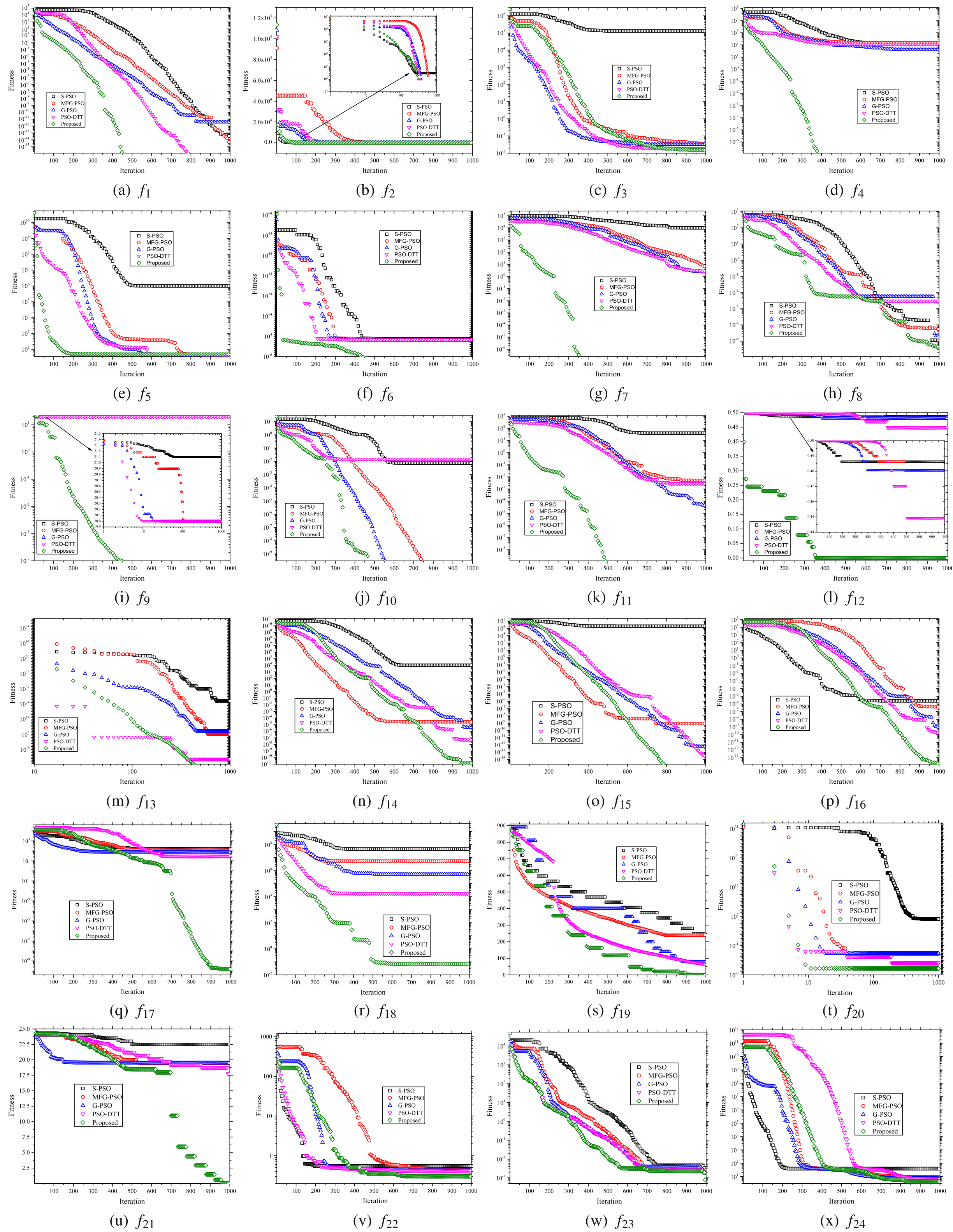
Step 1: Input the fitness function of optimal problem $f(X)$, the maximum iteration *Max_Iter*, the position limit of particles *Max_Iter* and the damping factor α.

Step 2: Initiate PSO-DFCM by generating the position and velocity of all the particles, calculate the fitness of each particle, and set the current fitness value and position as the historical best value and the historical best position. Find the global best value and the global best position.

Table 1
Non-constrained 24 benchmark functions.

Function	Form	Geometry shape	Separable	Other features
Sphere	$f_1 = \sum_{i=1}^{D} x_i^2$	Unimodal	No	Easy to Converge
Noncontiguous-sphere	$f_2 = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	Unimodal	Yes	Easy to Converge
Uniform-distribution perturbed ellipsoid	$f_3 = \sum_{i=1}^{D} i x_i^2 + U(0, \delta)$	Unimodal	Yes	Affected by Perturbing Term
Rastrigrin	$f_4 = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	Multimodal	No	Large Number of Local optima
Rosenbrock	$f_5 = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	Unimodal	No	The Global Optimum in a Long Narrow Parabolic-Shaped-Flat Valley
Schwefel	$f_6 = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	Unimodal	Yes	Hard to Converge to the Global Optimum in Some Directions
1st Schwefel	$f_7 = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	Unimodal	No	Symmetric & Great Gradient Difference in Adjacent Directions
2nd Schwefel	$f_8 = \sum_{i=1}^{D} \lvert x_i \rvert \lvert \sin(\sqrt{\lvert x_i \rvert}) \rvert$	Multimodal	Yes	Large Number of Local optima
Ackley	$f_9 = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$	Multimodal	No	a Nearly Flat Outer Region
Griewank	$f_{10} = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}) + 1$	Multimodal	No	Easily Trapped in Local Optima
Alpine	$f_{11} = \sum_{i=1}^{D} (\lvert x_i \sin(x_i) \rvert + 0.1 x_i)$	Unimodal	No	Easily Premature over a Few Iterations
Schaffer	$f_{12} = 0.5 + \frac{\sin^2(\sqrt{\sum_{i=1}^{D} x_i^2}) - 0.5}{1 + 0.1\sum_{i=1}^{D} x_i^2}$	Multimodal	No	Large Number of Local optima
Different power	$f_{13} = \sum_{i=1}^{D} \lvert x_i \rvert^{i+1}$	Unimodal	No	Asymmetric
Bent cigar	$f_{14} = x_1^2 + 10^6\sum_{i=2}^{D} x_i^2$	Unimodal	No	Smooth but Narrow Ridge
Discus	$f_{15} = 10^6 x_1^2 + \sum_{i=2}^{D} x_i^2$	Unimodal	No	Sharp but Broad Ridge
Zakharov	$f_{16} = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5 x_i)^2 + (\sum_{i=1}^{D} 0.5 x_i)^4$	Unimodal	No	Dramatic Gradient Discrepancy at Different Directions
Levy	$f_{17} = \sin^2(\pi w_1) + \sum_{i=1}^{D-1} (w_i - 1)^2(1 + 10\sin^2(\pi w_i + 1)) + (w_D - 1)^2(1 + \sin^2(2\pi w_D))$	Multimodal	No	Large Number of Local optima
High conditioned elliptic	$f_{18} = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i^2$	Unimodal	No	The Global Optimum in a Long Smooth and Wide Valley
Weierstrass	$f_{19} = \sum_{i=1}^{D} (\sum_{k=0}^{kmax} (a^k \cos(2\pi b^k(x_i + 0.5)))) - D\sum_{k=0}^{kmax} a^k \cos(\pi b^k)$	Multimodal	No	Continuous Everywhere but Differentiable Nowhere
Katsuura	$f_{20} = \frac{10}{D^2}\prod_{i=1}^{D} (1 + i\sum_{j=1}^{32} \frac{\lvert 2^j x_i - round(2^j x_i) \rvert}{2^j})^{\frac{10}{D^{1.2}}} - \frac{10}{D^2}$	Multimodal	Yes	Many Similar Local Optima to the Global Optimum
Expanded Schaffer	$f_{21} = f_{12}(x_1, x_2) + f_{12}(x_2, x_3) + \cdots + f_{12}(x_{D-1}, x_D) + f_{12}(x_D, x_1)$	Multimodal	No	Similar to Schaffer but Local Optima Closer to Each Other
HappyCat	$f_{22} = \lvert \sum_{i=1}^{D} x_i^2 - D \rvert^{\frac{1}{4}} + \frac{\frac{1}{2}\sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i}{D} + \frac{1}{2}$	Multimodal	No	the Better Optimum the Narrower Region
HGBat	$f_{23} = \lvert (\sum_{i=1}^{D} x_i^2)^2 - (\sum_{i=1}^{D} x_i)^2 \rvert^{\frac{1}{2}} + \frac{\frac{1}{2}\sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i}{D} + \frac{1}{2}$	Multimodal	No	Different Properties around Different Local Optima
Expanded Griewank	$f_{24} = f_{10}(f_5(x_1, x_2)) + f_{10}(f_5(x_2, x_3)) + \ldots + f_{10}(f_5(x_{D-1}, x_D)) + f_{10}(f_5(x_D, x_1))$	Multimodal	No	

Where $w_i = 1 + \frac{x_i - 1}{4} (\forall i = 1, \ldots, D)$, $a = \frac{1}{2}$, $b = 3$, $kmax = 20$.



(a) f_1 (b) f_2 (c) f_3 (d) f_4
(e) f_5 (f) f_6 (g) f_7 (h) f_8
(i) f_9 (j) f_{10} (k) f_{11} (l) f_{12}
(m) f_{13} (n) f_{14} (o) f_{15} (p) f_{16}
(q) f_{17} (r) f_{18} (s) f_{19} (t) f_{20}
(u) f_{21} (v) f_{22} (w) f_{23} (x) f_{24}

Fig. 1. Fitness of 24 benchmark functions in 50D Space over Iterations.

Step 3: Update the global best value and the temporary global best position using Eq. (4) and Eq. (8a), respectively.
Step 4: Use fuzzy c-means clustering to cluster the particles' positions for the neighborhood establishment. Update the local best value and the temporary local best position using Eq. (6) and Eq. (8b), respectively.
Step 5: Start cooperation between local and global swarms using Eq. (9) to update global positions of all particles.
Step 6: Use Eq. (10) to carry out a weak particle reinitializing tactic to guarantee particles moving towards optimal direction as much as possible.
Step 7: Update positions and velocities of all particles.

Table 2
The optimal values of proposed PSO and compared PSOs (24 tests under 50 dimensions).

Function (50D)	S-PSO	MFG-PSO	G-PSO	PSO-DTT	Proposed
f_1	3.4849×10^{-12}	3.2832×10^{-14}	3.4733×10^{-15}	1.4121×10^{-21}	2.4544×10^{-88}
f_2	3	0	0	0	0
f_3	15.3814	0.0364	0.0318	0.0193	0.0149
f_4	151.23	150.24	116.41	39.798	1.1369×10^{-13}
f_5	78.109	47.83	45.686	41.452	0.8837
f_6	500.05	300.069	200.01	1.0165	1.4737×10^{-6}
f_7	1477.4	5.3609	2.8252	2.1165	9.7048×10^{-6}
f_8	2.9779×10^{-3}	5.5373×10^{-5}	2.0818×10^{-5}	1.0266×10^{-5}	8.7695×10^{-8}
f_9	21.101	20.5678	20.1276	1.2789	3.8348×10^{-6}
f_{10}	2.7961	1.4772×10^{-2}	7.3960×10^{-10}	1.4772×10^{-10}	2.2204×10^{-16}
f_{11}	40.6410	5.1119×10^{-3}	2.8313×10^{-3}	4.5049×10^{-5}	7.7318×10^{-8}
f_{12}	0.49743	0.48629	0.48049	0.47014	2.8008×10^{-3}
f_{13}	1.3438×10^{32}	1.0×10^{3}	3.6450×10^{3}	1.4912×10^{-8}	5.7234×10^{-13}
f_{14}	1.0×10^{4}	3.4459×10^{-6}	2.6637×10^{-5}	4.7793×10^{-8}	1.3637×10^{-11}
f_{15}	2.0×10^{4}	8.5719×10^{-10}	5.5301×10^{-13}	2.3545×10^{-14}	3.2578×10^{-19}
f_{16}	2.5252×10^{-6}	4.4133×10^{-7}	7.4766×10^{-10}	1.7879×10^{-10}	1.7150×10^{-14}
f_{17}	162.86	112.8	29.898	73.816	1.4779×10^{-10}
f_{18}	4.4776×10^{7}	5.2182×10^{6}	5.1795×10^{5}	1.0182×10^{5}	0.069413
f_{19}	249.23	239.69	79.2256	63.537	0.0252
f_{20}	1.8186×10^{6}	5.918×10^{-4}	6.8206×10^{-4}	1.8206×10^{-6}	5.8900×10^{-8}
f_{21}	22.4890	19.4820	19.4671	17.7185	1.1150×10^{-3}
f_{22}	0.54776	0.44299	0.41453	0.40875	0.30206
f_{23}	2.1700×10^{-3}	2.0500×10^{-3}	1.8900×10^{-3}	1.2101×10^{-3}	6.7315×10^{-4}
f_{24}	39.0842	8.9081	6.9491	6.3789	4.2696

Step 8: If the optimal value meets our requirement or the maximum iteration is reached, terminate the algorithm. Otherwise, go toStep 3 for a new iteration.

The pseudo code of the PSO-DFCM is presented in algorithm 1:

Algorithm 1: The Pseudo Code of Proposed PSO

Require: the fitness function of optimal problem $f(X)$, the maximum iteration *Max_Iter*, the position limit of particles X_{max} and the damping factor α
Initialize: $iter = 0, X_j^{iter} \sim U(-X_{max}, X_{max})$ and $V_j^{iter} = 0$
Stop Criterion: $iter \leq Max\_Iter$ or the optimal value is not reached
Update ${}^{g}V_j^{iter+1}$ in conformity with Eq. (4).
Move particles' global transient positions according to Eq. (8a).
Update ${}^{l}V_j^{iter+1}$ in conformity with Eq. (6).
Move particles' local transient positions according to Eq. (8b).
Limit the particles' global and local transient positions.
Start cooperation between local and global swarms based on Eq. (9).
Abandon weak particles and reinitialize their positions using Eq. (10).
Update particles' *pbest*, $gbest_l$, $gbest_g$ and *pleast* information.
$iter \leftarrow iter + 1$

3. Experiments and results analysis

In order to test the performance of our proposed method, 24 benchmark functions are adopted from the paper [39] to validate our proposal algorithm. The benchmark functions in Table 1 have either a narrow valley, basin, or a huge number of local optima, which are challenging for optima-search algorithms. In the comparative study with the standard PSO and three state-of-art variants of PSO, it is demonstrated that the proposed method is more adaptive to large scale and high-dimensional searching optimal problems:

(1) Standard PSO (S-PSO) [4]
(2) Multi-function Global PSO (MFG-PSO) [18]
(3) Gravitational PSO (G-PSO) [20]
(4) PSO Using Dynamic Tournament Topology (PSO-DTT) [40]

In order to compare our proposed PSO with other PSO variants, dimensions of all tested benchmark functions are set to 50; the maximum iteration step of each improved PSO is fixed at 1×10^3, and the searching space is $-100 \sim 100$ in all non-constrained benchmark functions. Table 1 lists 24 on-constrained benchmark functions. Tables 2–4 show the optimal values of the proposed PSO and compared PSOs under 50, 30 and 10 dimensions, respectively. Fig. 1 demonstrates fitness of 24 benchmark functions in 50-dimensional space over iterations with the proposed PSO and compared PSOs.

For unimodal geometrical functions ($f_{1-3,5-7,11,13-16}$), one can observe that our proposal's performance outperforms four other algorithms in term of final optimal values. But for convergence speed, the proposed method is the fastest in $f_{1-2,5-6,11}$ but not in $f_{3,13-16}$. As to multimodal geometrical functions ($f_{4,8-10,12}$), the suggested method has an efficient capacity of escaping from the local optima points since the re-initialized positions for abandoned particles contribute to diversity increase among swarms.

In the comparative study shown in Tables 2–4, it is easily observed that reaching the tolerated ranges of optimal values needs more computational efforts over the rise in problem dimensions. When local optima and global optimum in objective problems are highly huddled together in high-dimensional space such like f_{12}, finding the potential global optimal points is an arduous task for computers; some algorithms cannot tackle well the challenges in large barriers and local optima traps when approaching the global optimal points. From the comparative results shown in Figs. 1(a)–1(p) and Tables 2–4, our proposed PSO has better performance in finding and converging to the global optimal points, especially for high-dimensional optimal problems.

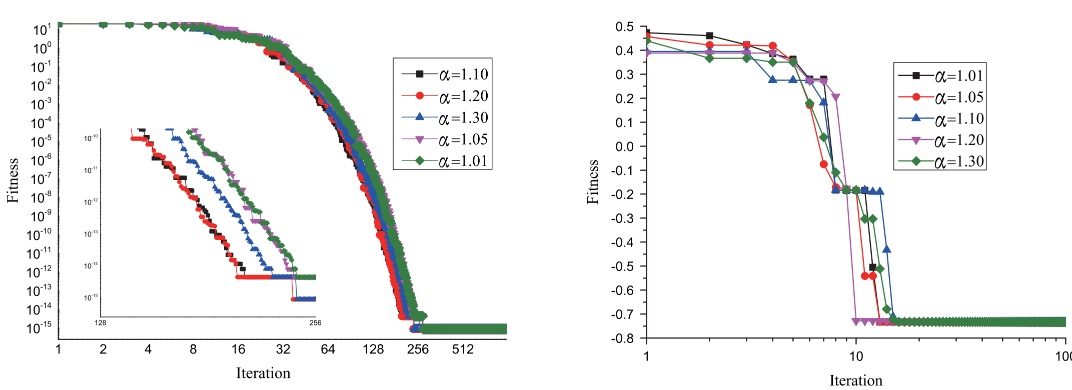
A new parameter, a damping factor α, is introduced in the proposed PSO, which is manually input by users based on prior information. Therefore, it is imperative to study how α affects the performance of our improved PSO, mainly on the convergence speed and the final global optimal value and its solutions. One adopts the functions of *Ackley* and *Schaffer* with 30 variables to find out the recommended values of α. From Figs. 2(a) and 2(b), one can observe that different α values have different effects on various optimal problems and $\alpha \in [1.0, 1.2]$ is recommended for most optimal problems.

Table 3
The optimal values of proposed PSO and compared PSOs (24 tests under 30 dimensions).

Function (30D)	S-PSO	MFG-PSO	G-PSO	PSO-DTT	Proposed
f_1	3.4407×10^{-27}	8.0012×10^{-27}	5.3766×10^{-29}	4.5759×10^{-30}	1.0272×10^{-38}
f_2	0	0	0	0	0
f_3	0.0149	0.0075	0.0078	0.0059	0.0021
f_4	73.6270	41.788	21.8891	0	1.1997×10^{-15}
f_5	16.126	16.008	11.536	17.993	0.2639
f_6	500	300	0.54216	3.3967×10^{-5}	1.3276×10^{-9}
f_7	1.8186×10^{-3}	2.6025×10^{-6}	1.0513×10^{-6}	1.2530×10^{-6}	2.7609×10^{-7}
f_8	1.1802×10^{-5}	1.296×10^{-9}	5.4424×10^{-10}	9.3069×10^{-14}	3.7774×10^{-14}
f_9	20	20	20	2.66×10^{-15}	8.8817×10^{-8}
f_{10}	0.0132	0.0074	0.0369	1.1102×10^{-16}	0
f_{11}	0.16379	4.6163×10^{-7}	5.7395×10^{-7}	1.7238×10^{-7}	1.1705×10^{-11}
f_{12}	0.39813	0.44918	0.39813	0.44918	2.4520×10^{-4}
f_{13}	4.6582×10^{21}	10.1253	9.9153	6.3246×10^{-13}	8.5617×10^{-17}
f_{14}	1.0×10^{4}	1.1584×10^{-21}	1.0000×10^{-18}	1.4076×10^{-23}	2.762×10^{-26}
f_{15}	3.5337×10^{-22}	2.3683×10^{-29}	1.3158×10^{-25}	3.1181×10^{-30}	1.4563×10^{-40}
f_{16}	2.1575×10^{-22}	2.3491×10^{-22}	3.1189×10^{-25}	1.6514×10^{-27}	2.0309×10^{-35}
f_{17}	114.89	0.9897	5.6708×10^{-25}	1.2198×10^{-27}	3.0815×10^{-33}
f_{18}	1.1615×10^{6}	7.6993×10^{5}	1.1826×10^{5}	8.5210×10^{-10}	5.8791×10^{-25}
f_{19}	142.2310	113.46	28.1026	13.5372	0.0072
f_{20}	0.68195	0	0	0	0
f_{21}	11.994	11.705	11.492	11.492	2.7153×10^{-4}
f_{22}	0.51693	0.43708	0.25112	0.24713	0.21061
f_{23}	6.7579×10^{-4}	4.5662×10^{-4}	3.7527×10^{-4}	3.0054×10^{-4}	2.3162×10^{-6}
f_{24}	14.9267	5.1723	2.8651	2.7687	1.4319

Table 4
The optimal values of proposed PSO and compared PSOs (24 tests under 10 dimensions).

Function (10D)	S-PSO	MFG-PSO	G-PSO	PSO-DTT	Proposed
f_1	1.2981×10^{-33}	2.6476×10^{-67}	1.4636×10^{-71}	2.2207×10^{-75}	7.5135×10^{-100}
f_2	0	0	0	0	0
f_3	0.0134	0.0018	0.0012	0.0031	0.0002
f_4	4.9748	2.9849	2.9849	0	0
f_5	0.7236	0.0313	0.1961	0.067817	1.0174×10^{-6}
f_6	1.6792×10^{-31}	1.8431×10^{-34}	3.7758×10^{-35}	2.3531×10^{-37}	3.0834×10^{-45}
f_7	2.2444×10^{-33}	5.4982×10^{-45}	3.5619×10^{-47}	1.9476×10^{-50}	4.353×10^{-55}
f_8	6.2851×10^{-14}	7.2521×10^{-15}	3.6261×10^{-15}	1.4504×10^{-14}	3.6259×10^{-15}
f_9	20	20	20	0	6.6529×10^{-11}
f_{10}	0.017226	0.095866	0.051657	0	0
f_{11}	2.2704×10^{-14}	2.3315×10^{-15}	3.8858×10^{-15}	6.1062×10^{-16}	2.2204×10^{-16}
f_{12}	0.2452	0.2452	0.2452	0.2452	1.0519×10^{-7}
f_{13}	5.1321×10^{-34}	6.2584×10^{-109}	6.4853×10^{-114}	4.3336×10^{-116}	9.7942×10^{-124}
f_{14}	1.0×10^{4}	3.8447×10^{-63}	4.2304×10^{-34}	2.8381×10^{-68}	6.4535×10^{-94}
f_{15}	1.9378×10^{-33}	2.5672×10^{-66}	4.8278×10^{-69}	1.9491×10^{-73}	3.3106×10^{-99}
f_{16}	5.6294×10^{-34}	7.5227×10^{-68}	2.9228×10^{-74}	8.0551×10^{-68}	5.7021×10^{-97}
f_{17}	0	0	0	0	0
f_{18}	1.0×10^{4}	0.0830	0.0010	0	0
f_{19}	52.1023	33.6416	7.4766	0.3153	0
f_{20}	0	0	0	0	0
f_{21}	3.4973	2.9974	1.4995	0.9989	0
f_{22}	6.5738×10^{-6}	4.8638×10^{-6}	4.0837×10^{-6}	1.7879×10^{-6}	0
f_{23}	0	0	0	0	0
f_{24}	2.0659	0.5816	0.3509	0.2861	0.0137



(a) *Ackley* (b) *Schaffer*

Fig. 2. Performance impact of different α.

4. Conclusions

In this work, a novel variant of PSO (PSO-DPCM) is proposed considering *damping factor* and *cooperative mechanism*. Two versions of collective best particles are aggregated together to take advantage of the merits of both. Two swarms are employed to find out optimal positions based on the cooperative mechanism. In the local-best-oriented swarm, random walk is also incorporated into particles' velocity updates as a perturbing term for the premature convergence avoidance of local best particles and the neighborhood mode of individuals is adaptively established by the application of *fuzzy c-means* clustering to particles' positions. To strengthen the exploring and exploiting competences of particles, a damping factor is introduced to adjust the position information inherited from the previous state. It is inevitable that some particles will become unfavorable and weak over evolutions. The proposed method also utilizes an abandoned tactic to reduce the negative effect of weak particles on optimal results. Comprehensive experiments have validated that our proposal has a good globally-searching capacity and performs effectively and reliably when compared with standard PSO and three state-of-art variants of PSOs.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (41274109), the Innovative Team Project of Sichuan Province, China (2015TD0020), and the Research Fund of State Key Laboratory of Geohazard Prevention and Geoenvironment Protection, China (SKLGP2016Z009).

References

[1] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, vol. 4, Piscataway, 1995, 1942–1948.
[2] K.K. Vardhini, T. Sitamahalakshmi, A review on nature-based swarm intelligence optimization techniques and its current research directions, Indian J. Sci. Technol. 9 (10) (2016).
[3] R. Wang, W. Ji, M. Liu, X. Wang, J. Weng, S. Deng, S. Gao, C. Yuan, Review on mining data from multiple data sources, Pattern Recognit. Lett. 109 (2018) 120–126.
[4] B. Xue, M. Zhang, W. Browne, Particle swarm optimisation for feature selection in classification: a multi-objective approach, IEEE Trans. Cybern. 43 (6) (2013) 1656–1671.
[5] A. Khare, S. Rangnekar, A review of particle swarm optimization and its applications in solar photovoltaic system, Appl. Soft Comput. 13 (2013) 2997–3006.
[6] W.N. Chen, J. Zhang, Y. Lin, N. Chen, Z.H. Zhan, H.S.H. Chung, Y. Li, Y.H. Shi, Particle swarm optimization with an aging leader and challengers, IEEE Trans. Evol. Comput. 17 (2013) 241–258.
[7] R. Cheng, C.J. Yao, A social learning particle swarm optimization algorithm for scalable optimization, Inform. Sci. 291 (2015) 43–60.
[8] D.B.F. Van, A.P. Engelbrecht, A cooperative approach to particle swarm optimization, IEEE Trans. Evol. Comput. 8 (2004) 225–239.
[9] W. Elshamy, M.E. Hassan, B. Ahmed, Clubs-based particle swarm optimization, in: Swarm Intelligence Symposium SIS, vol. 2007, 2007, 289–296.
[10] L. Wang, Z. Cui, J. Zeng, Particle swarm optimization with group decision making, in: Ninth International Conference on Hybrid Intelligent Systems, 2009, pp. 388–393.
[11] M.S. Voss, Principal component particle swarm optimization, IEEE Congress Evol. Comput. 1 (2005) 298–305.
[12] J.H. Seo, C.H. Im, C.G. Heo, J.K. Kim, H.K. Jung, C.G. Lee, Multimodal function optimization based on particle swarm optimization, IEEE Trans. Magn. 42 (4) (2006) 1095–1098.
[13] J. Tang, X. Zhao, A fine tuning hybrid particle swarm optimization algorithm, in: International Conference on Future Biomedical Information Engineering, 2009, pp. 296–299.
[14] L. Han, X.S. He, A novel opposition-based particle swarm optimization for noisy problems, in: 2007 IEEE Third International Conference on Natural Computation, vol. 3, 2007, pp. 624–629.
[15] L.M. Zhang, Y. Tang, C. Hua, X. Guan, A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques, Appl. Soft Comput. 28 (2015) 138–149.
[16] Y.P. Chen, W.C. Peng, M.C. Jian, Particle swarm optimization with recombination and dynamic linkage discovery, IEEE Trans. Syst. Man Cybern. B 37 (2007) 1460–1470.
[17] F.T.M. Silva, B.A. Pimentel, R.M.C.R. Souza, L.I.O. Adriano, Hybrid methods for fuzzy clustering based on fuzzy c-means and improved particle swarm optimization, Expert Syst. Appl. 42 (2015) 6315–6328.
[18] R.H. Zhao, Y. Yuan, Q.X. Chen, C.X. Zhang, Y. Shuai, H.P. Tan, A new multi-function global particle swarm optimization, Appl. Soft Comput. 49 (2016) 279–291.
[19] Y.Y. Hong, A.A. Beltran, A.C. Paglinawan, A chaos-enhanced particle swarm optimization with adaptive parameters and its application in maximum power point tracking, Math. Probl. Eng. 216 (2016).
[20] A. Yadav, K. Deep, J.H. Kim, A.K. Nagar, Gravitational swarm optimizer for global optimization, Swarm Evol. Comput. 31 (2016) 64–89.
[21] G. Pan, Q. Dou, X. Liu, Performance of two improved particle swarm optimizations in dynamic optimization environments, in: Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications, 2006, pp. 1024–1028.
[22] M.G. Epitropakis, V.P. Plagianakos, M.N. Vrahatis, Evolving cognitive and social experience in particle swarm optimization through differential evolution, Inform. Sci. 216 (2012) 50–92.
[23] J.S. Wu, L. Jin, M.Z. Liu, Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm, Neurocomputing 148 (2015) 136–142.
[24] C.F. Wang, K. Liu, P.P. Shen, Hybrid artificial bee colony algorithm and particle swarm search for global optimization, Math. Probl. Eng. 24 (2014) 1–9.
[25] H.D. Chen, S.Z. Wang, J.X. Li, Y. Li, A hybrid of artificial fish swarm algorithm and particle swarm optimization for feedforward neural network training, Int. J. Comput. Intell. Syst. (2007).
[26] B. Xue, M. Zhang, W. Browne, Particle swarm optimisation for feature selection in classification: novel initialisation and updating mechanisms, Appl. Soft Comput. 18 (2014) 261–276.
[27] H.T. Ye, W.G. Luo, Z.Q. Li, Convergence analysis of particle swarm optimizer and its improved algorithm based on velocity differential evolution, Comput. Intell. Neurosci. 2013 (2013).
[28] J. Chen, F. Pan, T. Cai, Acceleration factor harmonious particle swarm optimizer, Int. J. Autom. Comput. 1 (2006) 41–46.
[29] C.W. Cleghorn, A.P. Engelbrecht, Particle swarm variants: standardized convergence analysis, Swarm Intell. 9 (2015) 177–203.
[30] D.P. Tian, A review of convergence analysis of particle swarm optimization, Int. J. Grid Distrib. Comput. 6 (6) (2013) 117–128.
[31] S.S. Aote, M.M. Raghuwanshi, L. Malik, A brief review on particle swarm optimization: limitations & future directions, Int. J. Comput. Sci. Eng. 2 (5) (2013) 196–200.
[32] R.S. Na, Philosophy of Science and Technology, Peking University Press, Beijing, 2006.
[33] J. Jones, Cooperative foraging in the mountain caracara in peru, Wilson Bull. 111 (3) (1999) 437–439.
[34] S. Eliassen, C. Jøgensen, Extra-pair mating and evolution of cooperative neighbourhoods, Plos One 9 (4) (2014) e99878.
[35] X.P. Wei, J.X. Zhang, D.S. Zhou, Q. Zhang, Multiswarm particle swarm optimization with transfer of the best particle, Comput. Intell. Neurosci. 2015 (2015).
[36] M. Hasanzadeh, M.R. Meybodi, M.M. Ebadzadeh, A learning automata approach to cooperative particle swarm optimizer, J. Inf. Syst. Telecommun. 2 (1) (2014) 1–14.
[37] F. van den Bergh, A.P. Engelbrecht, A cooperative approach to particle swarm optimization, IEEE Congress Evol. Comput. 8 (3) (2004) 225–239.
[38] S.S. Jadon, H. Sharma, J.C. Bansal, R. Tiwari, Self adaptive acceleration factor in particle swarm optimization, in: Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications, 2012, pp. 325–340.
[39] B.Y. Qu, J.J. Liang, P.N. Suganthan, Q. Chen, Problem definitions and evaluation criteria for the CEC 2017 competition on single objective multi-niche optimization, Technical Report, 2017.
[40] L. Wang, B. Yang, J. Orchard, Particle swarm optimization using dynamic tournament topology, Appl. Soft Comput. 48 (2016) 584–596.