Niching Particle Swarm Optimizer with Entropy-Based Exploration Strategy for Global Optimization

Dongyang Li[1], Weian Guo[2(✉)], and Lei Wang[1]

[1] Department of Electronics and Information Engineering, Tongji University, Shanghai 201804, China
lidongyang0412@163.com, wangle_tj@126.com
[2] Sino-German College of Applied Sciences, Tongji University, Shanghai 201804, China
guoweian@tongji.edu.cn

Abstract. As a kind of evolutionary algorithms, particle swarm optimization is famous for its simplicity and efficiency in optimization. However, for complex problems, PSO is prone to be trapped into the local optima. To address this issue, a particle swarm optimizer with niching strategy and entropy-based exploration strategy (PSO-NE) is proposed in this paper. To be specific, on one hand, a distance based niching strategy and the competitive learning strategy are adopted to design the exploitation in PSO-NE; on the other hand, the exploration in PSO-NE is achieved by an entropy based exploring strategy. With such kind of designs, the exploitation and exploration in PSO-NE can be dependently adjusted, which is beneficial for balancing these two factors. To validate the effectiveness of the proposed algorithm, extensive experiments have been conducted based on 28 benchmarks from CEC' 2013. The proposed algorithm shows its competitive performance with comparing to six other typical variants of PSO.

Keywords: Particle swarm optimizer · Global optimization · Niching strategy · Competitive learning · Entropy

1 Introduction

As a kind of evolutionary algorithms (EAs), particle swarm optimization (PSO) has attracted a huge amount of attentions since its advent [1, 2]. The basic PSO keeps a number of particles each of which includes two attributes, the velocity and position which are iteratively updated according to the following equations

$$v_i^d(t+1) = w \cdot v_i^d(t) + c_1 \cdot r_1 \cdot \left(pbest_i^d(t) - p_i^d(t)\right) + c_2 \cdot r_2 \cdot \left(gbest^d(t) - p_i^d(t)\right) \quad (1)$$

$$p_i^d(t+1) = p_i^d(t) + v_i^d(t+1) \quad (2)$$

where t is the generation number; p_i^d and v_i^d are the dth dimensions of the position and velocity of ith particle respectively; $pbest_i^d$ represents the dth dimension of the best position searched by ith particle in current generation while $gbest^d$ are the best position

Y. Tan et al. (Eds.): ICSI 2019, LNCS 11655, pp. 118–127, 2019.
https://doi.org/10.1007/978-3-030-26369-0_11

of the whole swarm in current generation; w is the inertia weight, c_1 and c_2 are the acceleration coefficients; r_1 and r_2 are two randomly generated number within $[0, 1]$. Due to its simplicity and efficiency, PSO has been widely used in many studies areas, such as antenna designs [3], feature selection [4], robot path planning [5], and power system [6].

But many researchers have found that PSO lacks efficiency in solving complex optimizations. Thus, a lot of studies can be found in past two decades. First, many researchers focus on the adjustment of the parameters in PSO including the deterministic control methods [7–9] and the adaptive control strategies [10, 11]. Second, some efforts are made to diversify the exemplars for particles, such as the fully informed particle swarm optimization [12], the comprehensive learning particle swarm optimizer [13], and the competitive swarm optimizer (CSO) [17]. Besides, hybridization with other techniques is also much accounted by many studies such as particle swarm optimizer with crossover [18] and cooperatively coevolutionary strategy [19].

However, PSO and its variants still always fail to find the global optima in many cases. That's mainly because the existing PSO variants usually can't get a good balance between exploitation and exploration. For example, CSO enhances the exploration by learning the mean position of the whole swarm. However, such mean position is shared by all the particles to be updated [17], which results in a highly coupled relationship between exploration and exploitation in such kind of learning strategy.

To address this issue, this paper aims to reduce the coupling between exploitation and exploration in PSO and the main contributions in this paper are in following.

(1) A niching strategy is adopted to classify the particles into several sub-groups based on the distances among particles; then, a competitive learning strategy is employed to exploit each sub-group.
(2) An entropy based exploring strategy is proposed to enhance the exploration of PSO.

The rest of this paper is organized as follows. A brief overview of the related works on PSO will be presented in Sect. 2. In Sect. 3, the details of the proposed algorithm are introduced. The experiments and the results analyses are conducted in Sect. 4 and we end this paper with the conclusion in Sect. 5.

2 Related Works

For improving the performance of PSO, the existing works can be mainly categorized into following three classes.

2.1 Modified Updating Strategies

In such kind of variants, the main idea is to propose new learning strategy to improve the search ability of PSO. Mendes and Kennedy designed a fully informed learning strategy where particles learn to the contracted exemplars based on their neighbors [12]. Liang proposed comprehensive learning which allows each particle's *pbest* to be a leader [13]. Chen and Zhang proposed ALCPSO in which the *gbest* can be challenged and replaced by another generated particle in some cases leading to a more diverse global leader [20]. Cheng and Sun proposed FBE which includes two sub-swarms and a fitness value based competition. They identify the weak particles and the strong particles at each generation, then each weak particle is leaded by the best and another randomly selected particles from another sub-swarm while the strong particles are subjected to a mutation operation [14]. Later in 2015, Cheng and Jin proposed the competitive swarm optimizer which shows promising performance both in low dimensional and large scale optimization [17]. In the same year, Cheng and Jin proposed SL-PSO allowing particles to learn from each particles that better than themselves [15]. Last year, Yang and Chen proposed DLLSO in recently [16]. DLLSO grades particles base on the fitness values, and every particle chooses two different exemplars from superior levels. By this way, particles in DLLSO can get more diverse exemplars than CSO.

2.2 Parameter Control Strategies

The parameters are essential for controlling the convergence and diversity of PSO. Shi proposed a linear control methods and a fuzzy control strategy for the adjustment of the inertia weight [21, 22]. Ratnaweera and Halgamuge proposed HPSO-TVAC, where the acceleration coefficients are varying during the run of the algorithm [8]. APSO proposed by Zhan adjusts the acceleration coefficients according to an evolutionary state estimation strategy [10].

2.3 Hybridization with Other Techniques

The hybridization with other techniques is to utilize other techniques to enhance the search ability of the basic PSO. Such as the $CCPSO\text{-}S_K$ and $CCPSO\text{-}H_K$ integrated the co-operative co-evolutionary framework into PSO [23, 24]. Following this idea, Li and Yao put forward CCPSO2 in which the Gaussian and Cauchy distribution are used to update individuals to make a balance between exploitation and exploration [25]. Qin and Cheng proposed a PSO variant by dividing the whole swarm into learned and learning sub-swarms at each generation. And the learning sub-swarm will learn from the learned sub-swarm with a random probability [26]. The genetic learning particle swarm optimization put forward by Gong and Li uses crossover and mutation operators to enhance the exploration ability of PSO [27]. Similar to this idea, Chen and Li designed two different crossover operations to breed promising exemplars in their recent work [18].

More about the development of PSO can be found in [28]. Although these works have promoted PSO in various kinds of optimizations, PSO and its variants are still less effective for complex optimization. As discussed in Sect. 1, this is mainly because most of the existing methods still cannot achieve a good balance between exploitation and exploration, especially in solving problems with high complexity.

Thus, to further decouple the exploitation and exploration and get more reasonable balance between these two factors for PSO. This paper proposed a novel variant of PSO with introducing a niching method, the competitive learning strategies and an entropy-based exploration tactics. The details will be presented in the following section.

3 Proposed Algorithm

As discussed above, one can found that PSO still need to be further improved in balancing its exploitation and exploration. Thus, we proposed a novel variant of PSO as following.

3.1 Exploitation Operator

Exploitation is important for PSO because it can help the algorithm to refine the located promising areas. However, for the basic PSO, the current global best position of the swarm at each generation may be not in the most promising area in the search space under some cases during the optimization, especially for the multi-modal optimization. This can be illustrated by Fig. 1.

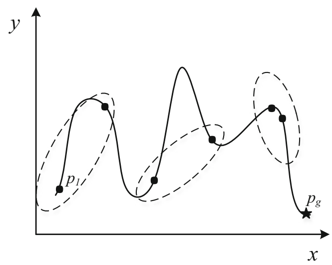

Fig. 1. A potential function with one decision variable.

As shown in Fig. 1, the true global best solution is p_g while the current global best position is p_1. And in the basic PSO, all the particle will be attracted by p_1 which will lead to missing of the true global best solution. To address this issue, it is better to group the swarm to different sub-groups as shown in Fig. 1. By this way, more promising solutions can be exploited which can potentially address the aforementioned issue. To achieve this, a niching method and the competitive learning strategy are adopted which will be introduced following.

Algorithm 1. Niching strategy: Clustering for Speciation [29]

Input : Population $\boldsymbol{P}$, cluster size $\boldsymbol{M}$

Step 1: Sort $\boldsymbol{P}$ according to fitness;

Step 2: While $\boldsymbol{P}$ is not empty

 Select the best individual $\boldsymbol{P_{best}}$ in $\boldsymbol{P}$ as a new seed;

 Build a species containing $\boldsymbol{P_{best}}$ and M-1 individuals nearest to it;

 Eliminate these $\boldsymbol{M}$ individuals from $\boldsymbol{P}$;

 End While

Output: A set of species

Then, in each group, for a thorough exploitation, the competitive learning strategy proposed in [17] is used to identify the particles to be updated and the corresponding exemplars. Consequently, the velocity of a particle to be updated in subgroup in the exploitation phase will be updated by using (3)

$$v_{m,l,i}^{d}(t+1) = r_1 \cdot \left(p_{m,w,i}^{d}(t) - p_{m,l,i}^{d}(t)\right) \tag{3}$$

where $v_{m,l,i}^{d}(t)$ and $p_{m,l,i}^{d}(t)$ denote the dth dimension of the ith particle to be updated in mth subgroup and the corresponding information of the exemplars; r_1 holds the same meaning with that in (1).

3.2 Exploration Operator

For exploration, a commonly used way is to keep the particles uniformly dispersed in the search space. To achieve this, this paper suggests an entropy based exploring strategy because entropy is an effective measure to evaluate the scatter degree of a set of random variables. Specifically, we adopt a modified entropy measurement based on the design proposed in [30], where the particles will be first sorted according to their fitness values and then the entropy for ith particle is defined as following

$$entropy_i = -normal\left\{\left(pro_{i,1}\log_2\left(pro_{i,1}\right) + pro_{i,2}\log_2\left(pro_{i,2}\right)\right)\right\} \cdot normal\left\{\frac{L_i}{L}\right\} \tag{4}$$

$$pro_{i,1} = \frac{fitness_i - fitness_{i-1}}{L_i} \tag{5}$$

$$pro_{i,2} = \frac{fitness_{i+1} - fitness_i}{L_i} \tag{6}$$

$$L_i = fitness_{i+1} - fitness_{i-1} \tag{7}$$

$$L = \max(fitness) - \min(fitness) \tag{8}$$

where $normal\{\vec{z}\}$ is a normalize operator by dividing each element in $\vec{z}$ by the maximum value of $\vec{z}$. From (4)–(8) one can find that the proposed method takes both the uniform degree and the search space of particles in fitness landscape, which results in a more reasonable measurement for the uniform degree of particles. However, it is obvious that this method cannot compute the entropy for the best and worst particles, in this paper, such kind of particles' entropy are set to 0. Due to the space limitation, the detailed analysis for the benefits of the entropy-based crowding measurement will be not presented here, one can find it in [30].

Besides, to further enhance the flexibility of the search behavior of particles, a social learning based strategy is embedded in the exploration operator. The velocity updating of ith particle in exploration phase can be expressed in (9)

$$v_i^d(t+1) = c_2 \cdot r_2 \cdot \left(p_j^d(t) - p_i^d(t)\right) \tag{9}$$

where p_j^d is the dth dimension of the jth particle that randomly selected in the set in which all the particles' entropy is larger than ith particle; c_2 and r_2 hold the same meaning with that in (1).

In summary, the proposed velocity updating strategy for ith particle in mth subgroup that to be updated can be shown in (10) and (11)

$$v_{m,l,i}^d(t+1) = w \cdot v_{m,l,i}^d(t) + r_1 \cdot \left(p_{m,w,i}^d(t) - p_{m,l,i}^d(t)\right) + c_2 \cdot r_2 \cdot \left(p_j^d(t) - p_{m,l,i}^d(t)\right) \tag{10}$$

$$p_{m,l,i}^d(t+1) = p_{m,l,i}^d(t) + v_{m,l,i}^d(t+1) \tag{11}$$

To this end, the pseudo code of the proposed PSO-NE is shown in Algorithm 2.

Algorithm 2. Proposed algorithm: PSO-NE

Input : Population ***P***, cluster size ***M***, parameter $\boldsymbol{c_2}$
Step 1: Swam initialization;
Step 2: Compute the ***fitness*** for ***P***;
Step 2: While terminal criterion is false
    Sort ***P*** according to ***fitness***
    Execute the niching strategy according to Algorithm 1;
    Computing the ***entropy*** for particles using (4)-(8);
    Update swarm using (10)-(11);
    Compute the ***fitness*** for ***P***;
  End While
Output: Best particle

4 Experiments and Discussions

4.1 Experiments Settings

To validate the performance of PSO-NE, 28 benchmark functions from CEC's 2013 are employed to conduct the experiments. More details about the benchmarks can be find in [31]. Six other popular variants of PSO are adopted to compare with PSO-NE including ALCPSO [20], CLPSO [13], CSO [17], HPSO-TVAC [8], and FIPS [12], DLLSO [16].

In the experiments, the dimensionality D of the functions is set to 50. Each algorithm has a population size N of 40. The maximum fitness evaluations *MaxFEs* is set to $10000 \cdot D$, the search range is [−100 100]. For parameter settings, the cluster size $\boldsymbol{M}$ in PSO-NE is set to 10 while the parameter c_2 is set to 1 to 0.4 during the run. For the other five algorithms, we adopt the default parameter settings as following. In ALCPSO, w varies from 0.9 to 0.4 while $c_1 = c_2 = 1.49$; In CLPSO, w also varies

Table 1. The experimental results with fitness evaluations of 5e5.

Function	Property	ALCPSO	CLPSO	CSO	HPSO-TVAC	LIPS	DLLSO	PSO-NE
1	mean	3.19E-12	3.18E-13	2.27E-13	6.63E-12	6.79E-11	1.06E+01	2.27E-13
	pvalue	**8.68E-05**	**7.81E-03**	1.00E+00*	**8.82E-05**	**8.84E-05**	**8.86E-05**	-
2	mean	3.02E+07	3.53E+07	2.67E+06	7.71E+06	4.95E+07	5.01E+07	1.96E+06
	pvalue	**8.86E-05**	**8.86E-05**	**2.50E-03**	**8.86E-05**	**1.20E-04**	**8.86E-05**	-
3	mean	1.95E+09	3.65E+09	2.47E+09	2.33E+09	4.83E+09	2.83E+10	4.60E+05
	pvalue	**8.86E-05**	**8.86E-05**	**8.86E-05**	**8.86E-05**	**8.86E-05**	**8.86E-05**	-
4	mean	4.97E+03	3.42E+04	5.06E+04	2.40E+04	1.18E+05	1.29E+05	1.87E+03
	pvalue	**1.03E-04**	**8.86E-05**	**8.86E-05**	**8.86E-05**	**8.86E-05**	**8.86E-05**	-
5	mean	1.88E-12	3.87E-13	7.91E-11	1.33E-06	1.43E-08	1.48E+02	3.12E-13
	pvalue	**8.81E-05**	**2.07E-03**	**1.28E-03**	**8.86E-05**	**8.86E-05**	**8.86E-05**	-
6	mean	5.96E+01	4.62E+01	5.60E+01	9.28E+01	5.93E+01	1.59E+02	4.57E+01
	pvalue	**3.04E-02**	2.32E-01*	1.26E-01*	**2.93E-04**	**3.19E-03**	**8.86E-05**	-
7	mean	1.19E+02	1.19E+02	6.23E+01	1.75E+02	1.17E+02	1.06E+02	8.11E-01
	pvalue	**8.86E-05**	**8.86E-05**	**8.86E-05**	**8.86E-05**	**8.86E-05**	**8.86E-05**	-
8	mean	2.11E+01	2.11E+01	2.11E+01	2.11E+01	2.11E+01	2.11E+01	2.11E+01
	pvalue	8.23E-01*	1.91E-01*	7.37E-01*	6.27E-01*	3.32E-01*	7.37E-01*	-
9	mean	5.35E+01	5.39E+01	2.89E+01	5.83E+01	4.80E+01	3.94E+01	9.88E+00
	pvalue	**8.86E-05**	**8.86E-05**	**8.86E-05**	**8.86E-05**	**8.86E-05**	**8.86E-05**	-
10	mean	9.32E-01	1.47E+01	1.18E+00	2.41E+00	1.16E+00	3.30E+02	7.15E-02
	pvalue	**8.86E-05**	**8.86E-05**	**8.86E-05**	**8.86E-05**	**1.69E-02**	**8.86E-05**	-
11	mean	2.69E+01	7.39E-14	7.62E+01	8.11E+01	2.10E+02	1.37E+02	2.62E+01
	pvalue	9.70E-01*	8.84E-05	**8.86E-05**	**8.86E-05**	**8.86E-05**	**8.86E-05**	-
12	mean	3.17E+02	3.22E+02	7.68E+01	5.46E+02	2.40E+02	1.62E+02	1.71E+02
	pvalue	**1.63E-04**	**1.03E-04**	2.19E-04	**8.86E-05**	**1.71E-03**	3.70E-01*	-
13	mean	4.58E+02	4.13E+02	1.96E+02	6.78E+02	4.52E+02	3.77E+02	1.93E+02
	pvalue	**8.86E-05**	**8.86E-05**	6.81E-01*	**8.86E-05**	**8.86E-05**	**8.86E-05**	-
14	mean	1.04E+03	8.97E-01	2.13E+03	1.45E+03	4.64E+03	4.84E+03	1.05E+03
	pvalue	5.50E-01*	8.86E-05	**8.86E-05**	**3.19E-03**	**8.86E-05**	**8.86E-05**	-
15	mean	9.28E+03	8.25E+03	5.08E+03	8.40E+03	6.95E+03	6.91E+03	7.70E+03
	pvalue	**7.19E-03**	3.51E-01*	6.81E-04	1.91E-01*	2.18E-01*	1.17E-01*	-
16	mean	3.05E+00	1.89E+00	7.24E-01	2.75E+00	7.48E-01	5.66E-01	3.40E+00
	pvalue	3.59E-03	8.86E-05	8.86E-05	4.55E-03	8.86E-05	8.86E-05	-
17	mean	1.24E+02	5.08E+01	8.86E-05	2.57E+02	3.49E+02	2.11E+02	9.34E+01
	pvalue	**2.54E-04**	8.86E-05	**8.97E-03**	**8.86E-05**	**8.86E-05**	**8.86E-05**	-
18	mean	3.83E+02	4.07E+02	1.03E+02	8.21E+02	3.73E+02	2.79E+02	3.70E+02
	pvalue	5.50E-01*	**8.86E-05**	8.86E-05	**8.86E-05**	8.81E-01*	1.03E-04	-
19	mean	1.46E+01	5.47E-01	1.47E+01	1.94E+01	9.34E+01	6.00E+01	6.59E+00
	pvalue	**8.86E-05**	8.86E-05	**1.89E-04**	**8.86E-05**	**8.86E-05**	**8.86E-05**	-
20	mean	2.45E+01	2.36E+01	2.06E+01	2.24E+01	2.21E+01	2.34E+01	2.03E+01
	pvalue	**8.86E-05**	**8.86E-05**	6.81E-01*	**1.20E-04**	**2.19E-04**	**8.86E-05**	-
21	mean	7.74E+02	2.33E+02	6.18E+02	8.66E+02	3.78E+02	1.27E+03	9.36E+02
	pvalue	1.91E-01*	8.86E-05	6.90E-03	8.23E-01*	2.19E-04	**3.66E-02**	-
22	mean	2.68E+03	2.45E+01	2.65E+03	1.84E+03	6.68E+03	6.10E+03	1.18E+03
	pvalue	**8.86E-05**	**8.86E-05**	**8.86E-05**	**1.89E-04**	**8.86E-05**	**8.86E-05**	-
23	mean	9.78E+03	1.00E+04	5.74E+03	1.04E+04	9.07E+03	8.80E+03	8.56E+03
	pvalue	6.20E-02*	**4.00E-02**	1.71E-03	**3.04E-02**	5.02E-01*	7.37E-01*	-
24	mean	3.53E+02	3.51E+02	2.92E+02	3.81E+02	3.54E+02	3.22E+02	2.15E+02
	pvalue	**8.86E-05**	**8.86E-05**	**8.86E-05**	**8.86E-05**	**8.86E-05**	**8.86E-05**	-
25	mean	3.89E+02	3.90E+02	3.29E+02	3.77E+02	4.15E+02	3.60E+02	2.90E+02
	pvalue	**8.86E-05**	**8.86E-05**	**1.40E-04**	**8.86E-05**	**8.86E-05**	**8.86E-05**	-
26	mean	4.38E+02	2.04E+02	3.53E+02	4.22E+02	2.76E+02	4.00E+02	3.01E+02
	pvalue	**8.86E-05**	3.90E-04	**2.51E-02**	**5.17E-04**	2.47E-01*	**6.81E-04**	-
27	mean	1.74E+03	1.60E+03	1.18E+03	2.09E+03	1.74E+03	1.44E+03	5.19E+02
	pvalue	**8.86E-05**	**1.89E-04**	**8.86E-05**	**8.86E-05**	**8.86E-05**	**8.86E-05**	-
28	mean	2.14E+03	4.00E+02	7.25E+02	3.95E+03	8.76E+02	1.59E+03	4.00E+02
	pvalue	**8.84E-05**	**8.86E-05**	**2.98E-05**	**8.86E-05**	**8.68E-05**	**8.86E-05**	-
w/l/t		21/1/6	18/7/3	17/6/5	24/1/3	22/1/5	22/2/4	-

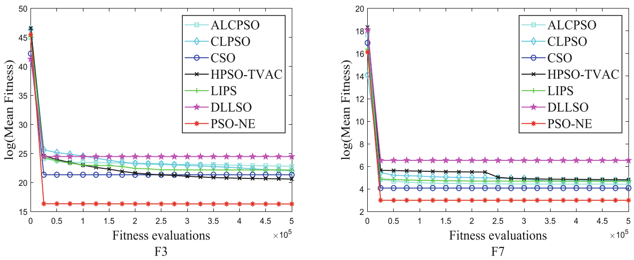

Fig. 2. The convergence figure on F3 and F7.

from 0.9 to 0.4, $c = 1.49445$ and $m = 7$; In CSO, *phi* is set to 0; In HPSO-TVAC, w varies from 0.9 to 0.4, c_1 varies from 0.5 to 2.5 while c_2 varies from 2.5 to 0.5; In FIPS, $\chi = 0.729$ and $\sum c_i = 0.41$; In DLLSO the level number set is $\{4, 6, 8, 10, 20\}$ and $\varphi = 0.4$. We run 20 simulations of each algorithm on each benchmark to get representative performance.

4.2 Results and Discussions

In this part, we record the difference between the obtained best fitness value and the real global optima in each run. Then the mean optimization results and the p-value obtained by Wilcoxon test are shown in Table 1, while Fig. 2 shows the convergence curve of these algorithms, due to the limitation of the space, we just draw the convergence figure on F_3 and F_7 which are unimodal and multimodal function respectively. In Table 1, the best results of mean performance are high lightened by gray; the p-value marked by bond and "*" indicate PSO-NE significantly better than and statistically equivalent to the compared algorithm on the corresponding function and. w/l/t at the bottom of the tables represent that how many times PSO-NE wins/loses/ties in the competitions with comparing to the corresponding algorithms.

From Table 1 we can see that the proposed PSO-NE wins for 16 functions with the mean fitness comparison. To be specific, for all the unimodal functions, PSO-NE obtains the best results, especially for F_2, F_3, PSO-NE performs much better than all the other algorithms; for the multi-modal functions, PSO-NE wins the first place on 6 functions; besides, the proposed PSO-NE performs competitive on the composition Functions as shown in Table 1.

For the p-value results, PSO-NE wins 21, 18, 17, 24, 22, 22 times over the corresponding algorithms, which stochastically demonstrates the competitive performance of PSO-NE. Additionally, the convergence figures also demonstrate the promising exploitation ability of PSO-NE.

In summary, the results shown in Table 1 turn out that the proposed strategy is effective for improving PSO in low dimension optimization.

5 Conclusions

In this paper, we propose a niching strategy and competitive learning based local exploitation strategy and an entropy based exploring strategy. On one hand, because the particles in PSO-NE can exploit local areas by cooperating with their neighbors, PSO-NE has a more reasonable exploitation ability; on the other hand, because of the designed entropy based exploring strategy, particles in PSO-NE perform better in exploration. Finally, 28 benchmarks are used to test the proposed algorithm with comparing to six other popular PSO variants. The results finally demonstrate the proposed algorithm is effective to deal with optimization problems.

Acknowledgments. This work was sponsored by the National Natural Science Foundation of China under Grant no. 71771176 and 61503287.

References

1. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: 6th Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43, IEEE, Nagoya (1995)
2. Eberhart, R., Kennedy, J.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE, Australia (1995)
3. Jin, N., Rahmat-Samii, Y.: Advances in particle swarm optimization for antenna designs: real-number, binary, single-objective and multiobjective implementations. IEEE Trans. Antennas Propag. **55**(3), 556–567 (2007)
4. Ghamisi, P., Benediktsson, J.A.: Feature selection based on hybridization of genetic algorithm and particle swarm optimization. IEEE Geosci. Remote Sens. Lett. **12**(2), 309–313 (2015)
5. Das, P.K., Behera, H.S., Panigrahi, B.K.: A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning. Swarm Evol. Comput. **28**, 14–28 (2016)
6. Koad, R.B.A., Zobaa, A.F., El-Shahat, A.: A novel MPPT algorithm based on particle swarm optimization for photovoltaic systems. IEEE Trans. Sustain. Energy **8**(2), 468–476 (2017)
7. Zheng, Y., Ma, L., Zhang, L., et al.: Empirical study of particle swarm optimizer with an increasing inertia weight. In: The 2003 Congress on Evolutionary Computation, vol. 1, pp. 221–226. IEEE, Canberra, ACT, Australia (2003)
8. Ratnaweera, A., Halgamuge, S.K., Watson, H.C.: Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. IEEE Trans. Evol. Comput. **8** (3), 240–255 (2004)
9. Chatterjee, A., Siarry, P.: Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. Comput. Oper. Res. **33**(3), 859–871 (2006)
10. Zhan, Z.H., Zhang, J., Li, Y., et al.: Adaptive particle swarm optimization. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **39**(6), 1362–1381 (2009)
11. Taherkhani, M., Safabakhsh, R.: A novel stability-based adaptive inertia weight for particle swarm optimization. Appl. Soft Comput. **38**, 281–295 (2016)

12. Mendes, R., Kennedy, J., Neves, J.: The fully informed particle swarm: simpler, maybe better. IEEE Trans. Evol. Comput. **8**(3), 204–210 (2004)
13. Liang, J.J., Qin, A.K., Suganthan, P.N., et al.: Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans. Evol. Comput. **10** (3), 281–295 (2006)
14. Cheng, R., Sun, C., Jin, Y.: A multi-swarm evolutionary framework based on a feedback mechanism. In: 2013 IEEE Congress on Evolutionary Computation, pp. 718–724. IEEE, Cancun (2001)
15. Cheng, R., Jin, Y.: A social learning particle swarm optimization algorithm for scalable optimization. Inf. Sci. **291**, 43–60 (2015)
16. Yang, Q., Chen, W.N., Da Deng, J., et al.: A level-based learning swarm optimizer for large-scale optimization. IEEE Trans. Evol. Comput. **22**(4), 578–594 (2018)
17. Cheng, R., Jin, Y.: A competitive swarm optimizer for large scale optimization. IEEE Trans. Cybern. **45**(2), 191–204 (2015)
18. Chen, Y., Li, L., Xiao, J., et al.: Particle swarm optimizer with crossover operation. Eng. Appl. Artif. Intell. **70**, 159–169 (2018)
19. Tang, R.L., Wu, Z., Fang, Y.J.: Adaptive multi-context cooperatively coevolving particle swarm optimization for large-scale problems. Soft. Comput. **21**(16), 4735–4754 (2017)
20. Chen, W.N., Zhang, J., Lin, Y., et al.: Particle swarm optimization with an aging leader and challengers. IEEE Trans. Evol. Comput. **17**(2), 241–258 (2013)
21. Shi, Y., Eberhart, R.C.: Empirical study of particle swarm optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation, vol. 3, pp. 1945–1950. IEEE, Washington (1999)
22. Shi, Y., Eberhart, R.C.: Fuzzy adaptive particle swarm optimization. In: Proceedings of the 2001 Congress on Evolutionary Computation, vol. 1, pp. 101–106. IEEE, South Korea (2001)
23. van den Bergh, F., Engelbrecht, A.: A cooperative approach to particle swarm optimization. IEEE Trans. Evol. Comput. **8**(3), 225–239 (2004)
24. den Bergh, F.V.: An analysis of particle swarm optimizers. Ph.D. dissertation, Dept. Comput. Sci., Univ. Pretoria, Pretoria, South Africa (2002)
25. Li, X., Yao, X.: Cooperatively coevolving particle swarms for large scale optimization. IEEE Trans. Evol. Comput. **16**(2), 210–224 (2012)
26. Qin, Q., Cheng, S., Zhang, Q., et al.: Particle swarm optimization with interswarm interactive learning strategy. IEEE Trans. Cybern. **46**(10), 2238–2251 (2016)
27. Gong, Y.J., Li, J.J., Zhou, Y., et al.: Genetic Learning Particle Swarm Optimization. IEEE Trans. Cybern. **46**(10), 2277–2290 (2017)
28. Bonyadi, M.R., Michalewicz, Z.: Particle swarm optimization for single objective continuous space problems: a review. Evol. Comput. **25**(1), 1–54 (2017)
29. Sheng, W., Swift, S., Zhang, L., Liu, X.: A weighted sum validity function for clustering with a hybrid niching genetic algorithm. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **35** (6), 1156–1167 (2005)
30. Wang, Y.N., Wu, L.H., Yuan, X.F.: Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure. Soft. Comput. **14**(3), 193–209 (2010)
31. Liang, J.J., Qu, B.Y., Suganthan, P.N., et al.: Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report, 201212(34): 281–295 (2013)