



A swarm optimizer with attention-based particle sampling and learning for large scale optimization

Mengmeng Sheng^{1,4} · Zidong Wang^{2,3} · Weibo Liu³ · Xi Wang¹ · Shengyong Chen⁵ · Xiaohui Liu³

Received: 17 May 2022 / Accepted: 14 September 2022 / Published online: 7 October 2022
© The Author(s) 2022

Abstract

Attention mechanism, which is a cognitive process of selectively concentrating on certain information while ignoring others, has been successfully employed in deep learning. In this paper, we introduce the attention mechanism into a particle swarm optimizer and propose an attention-based particle swarm optimizer (APSO) for large scale optimization. In the proposed method, the attention mechanism is introduced such that activating different particles to participate in evolution at different stages of evolution. Further, an attention-based particle learning is devised to randomly select three particles from a predominant sub-swarm, which is activated by the attention mechanism, to guide the learning of particles. The cooperation of these two strategies could be employed to achieve a balanced evolution search, thus appropriately searching the space of large-scale optimization problems. Extensive experiments have been carried out on CEC'2010 and CEC'2013 large scale optimization benchmark functions to evaluate the performance of proposed method and to compare with related methods. The results show the superiority of proposed method.

Keywords Particle swarm optimization · Attention mechanism · Exemplar selection · Large scale optimization

1 Introduction

Particle swarm optimization (PSO), proposed by Kennedy and Eberhart (1995), is an efficient swarm intelligence technology for global optimization. The underlying concept of PSO is to simulate the swarm behaviors of birds flocking. In PSO, the movements of particles are guided by their own best-known positions, denoted as *pBests*, as well as the best-known position of entire swarm, denoted as *gBest*, to search the space. PSO has been successfully applied in various fields (Gong et al. 2012; Ishibuchi and Salam 2013; Lu et al.

2021; Ji et al. 2021; Wang et al. 2021a). However, it may have difficulty to deal with optimization problems involving a large number of optima or high-dimensional data, which is termed as large scale global optimization (LSGO) (Yang and Pedersen 1997). Such an issue is mainly attributed to the premature convergence that usually occurs in the traditional PSO (Chen et al. 2013).

To address premature convergence, many variants of PSO have been proposed (Shi and Eberhart 1999; Zhan et al. 2009; Suganthan 1999; Kennedy and Mendes 2002; Juang 2004; Shelokar et al. 2007; Liang and Suganthan 2005; Li and Yao 2012; Cheng and Jin 2015a; Xu et al. 2021). One approach is to adaptively control the learning parameters of PSO. For example, Shi and Eberhart (1999) tried to linearly decrease the parameter value of inertia weight from 0.9 to 0.4 during evolution to balance exploration and exploitation. Zhan et al. (2009) designed a real-time evolutionary state estimation procedure to identify evolutionary state of the algorithm and automatically control the parameters of PSO based on the identified state. Another approach is to enhance the swarm diversity of PSO by introducing certain topological structures of swarm. For instance, in Suganthan (1999), the authors employed a neighborhood topology to enhance the diversity of swarm. Kennedy and Mendes (2002)

✉ Zidong Wang
zidong.wang@brunel.ac.uk

¹ School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 300023, China

² College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China

³ Department of Computer Science, Brunel University London, Uxbridge UB8 3PH, Middlesex, UK

⁴ Zhejiang Police College, Hangzhou 310053, China

⁵ School of Computer Science and Technology, Tianjin University of Technology, Tianjin 300382, China

proposed two topologies (i.e., a ring topology and a von Neumann topology) to enhance the diversity of evolutionary search. Further, since different search techniques possess different strengths, it is natural to hybridize PSO with other search methods, including other evolutionary algorithms (Juang 2004; Shelokar et al. 2007) and local searches (Liang and Suganthan 2005) to improve its performance. Additionally, since exchanging information between different sub-swarms can enhance the swarm diversity, a few multi-swarm based PSO methods have also been proposed. For example, Liang and Suganthan (2005) tried to dynamically re-group the swarm into multiple sub-swarms and exchange information among sub-swarms to achieve the diversity. Li and Yao (2012) presented a cooperative coevolutionary algorithm for LSGO. Apart from the above approaches, designing viable learning strategies to enhance the diversity of PSO has also gained attention. In this direction, for example, Cheng and Jin (2015a, b) devised several variants of PSO, in which particles are set to learn from predominant particles in the swarm. Such a predominant particle learning strategy is able to enhance the diversity of evolutionary search. Although the above PSO variants are able to improve the performance of traditional PSO, their capabilities to deliver a well-balanced evolutionary search could be limited.

In this paper, we propose an attention-based particle swarm optimizer (APSO) for LSGO. In cognitive science, human beings tend to selectively focus on some of visible information while ignoring other information (Itti et al. 1998; Corbetta and Shulman 2002) due to the bottleneck of information processing. The above mechanism is often referred to as the attention mechanism, which has been successfully employed in deep learning. The mechanism suggests that selectively paying attention to part of the information while ignoring other information can improve the efficiency of information processing. During evolution of PSO, different particles in the swarm generally have different potentials for searching the space and should be treated differently at different stages of evolution. In this sense, an attention-based particle sampling (APS) strategy, which works by adaptively activating partial particles to participate in evolution based on their fitness values and the stage of evolution, has been developed and incorporated into the proposed method. By paying attention to low quality particles at the early stage of evolution while gradually switching to high quality particles at the later stage of evolution, the APS can be used to support a well-balanced evolutionary search at swarm level. Moreover, an attention-based particle learning (APL) has also been devised to guide the learning of particles. The strategy works by selecting three particles randomly from a predominant sub-swarm, which is activated

by the attention mechanism, to participate into particle learning. By utilizing particles from the predominant sub-swarm activated by the attention mechanism for learning, the APL is able to enhance the balance of evolutionary search at particle level. Extensive experiments have been conducted on CEC'2010 (Tang et al. 2010) and CEC'2013 (Li et al. 2013) large scale benchmark sets to evaluate the significance of devised strategies and to compare the performance of proposed method with related algorithms. The results show our method is viable to deal with LSGO and outperforms related methods.

The rest of this paper is organized as follows. Section 2 gives a brief review of PSO and its variants as well as evolutionary algorithms dealing with LSGO. Section 3 presents the details of proposed method. Then, extensive experiments are conducted in Sect. 4 to verify the performance of proposed method. Finally, conclusions are given in Sect. 5.

2 Related work

2.1 PSO and its variants

PSO algorithm starts with a swarm of Np particles, which represent the solutions. To deal with an optimization problem, each particle i is defined as a position vector $X_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,D}\}$ associated with a velocity vector $V_i = \{v_{i,1}, v_{i,2}, \dots, v_{i,D}\}$, where D denotes the dimension of the problem. During evolution, the particles' own historical best positions will be recorded as $pBest_i = \{pBest_{i,1}, pBest_{i,2}, \dots, pBest_{i,D}\}$, $i = 1, 2, \dots, Np$, and the best one among these $pBests$ is denoted as $gBest$.

During particle learning, each particle i will update the velocity and position according to its $pBest_i$ and the $gBest$ as follows:

$$v_{i,j} = w \cdot v_{i,j} + c_1 \cdot r_{1,j} \cdot (pBest_{i,j} - x_{i,j}) + c_2 \cdot r_{2,j} \cdot (gBest_j - x_{i,j}) \quad (1)$$

$$x_{i,j} = x_{i,j} + v_{i,j}, \quad (2)$$

where w represents the inertia weight, r_1 and r_2 are random values between $[0, 1]$, c_1 and c_2 are acceleration coefficients and j denotes the j th dimension of the problem. After particle learning, if the current position X_i is better than $pBest_i$, then replace $pBest_i$ with X_i .

PSO has received much attention due to its efficiency and simplicity. Many studies have also been put forward to improve its performance, thereby deriving many variants of

PSO (Zhan et al. 2009; Kennedy and Mendes 2002; Liang and Suganthan 2005; Liang et al. 2006; Mendes et al. 2004). Among these variants, one approach is to focus on adaptively controlling of parameters of PSO. For example, in Shi and Eberhart (1998), the authors tried to improve the performance of PSO by linearly decreasing the inertia weight w along evolution according to rule:

$$w = w_{max} - (w_{max} - w_{min}) \cdot \frac{g}{G}, \quad (3)$$

where g represents the current number of generations and G is a predefined maximum number of generations. In Shi and Eberhart (2001), a fuzzy adaptive rule was proposed to set the value of w . While, a random setting of w was experimented in Eberhart and Shi (2001) for dynamic system optimization. In addition to inertia weight, acceleration coefficients c_1 and c_2 are also important for the performance of PSO. In Ratnaweera et al. (2004), the authors tried to linearly control acceleration coefficients in PSO to balance its global and local search capabilities. Zhan et al. (2009) tended to automatically control acceleration coefficients based on states of the PSO algorithm, which are identified by a real-time evolutionary state estimation procedure.

The second approach of enhancing traditional PSO is to introduce topological structures of swarm, thereby improving swarm diversity and alleviating premature convergence (Suganthan 1999; Kennedy 1999). For instance, Kennedy and Mendes (2002) proposed two topological structures (i.e., the ring topology and von Neumann topology) to organize the swarm. In this scheme, the updating rule of particle's velocity is defined as:

$$v_{ij} = w \cdot v_{ij} + c_1 \cdot r_{1j} \cdot (pBest_{ij} - x_{ij}) + c_2 \cdot r_{2j} \cdot (lBest_j - x_{ij}), \quad (4)$$

where $lBest_j$ denotes the best $pBest$ of i th particle's neighborhoods. In Mendes et al. (2004), the authors proposed a fully informed PSO, in which each particle is updated based on historical best positions of its neighbors. In Liang et al. (2006), the authors devised a comprehensive learning PSO, which allows the velocity of particle to be updated as:

$$v_{ij} = w \cdot v_{ij} + c \cdot r_j \cdot (pBest_{fi(j),j} - x_{ij}), \quad (5)$$

where $fi(j)$ denotes $pBest$ of the winner between two particles randomly selected from the swarm.

The third approach tends to improve PSO by combining other search operations or techniques. For example, Angelina (1998) introduced a tournament selection operation into PSO. Chen et al. (2007) and Andrews (2006) integrated the

crossover and mutation operation, respectively, from genetic algorithm (GA) into PSO. Juang (2004) tried to integrate a GA into PSO for designing a recurrent artificial neural network. Apart from GA, differential evolution (Zhang and Xie 2003), ant colony optimization (Shelokar et al. 2007; Song and Miao 2021) and local search (Liang and Suganthan 2005) have also been adopted to design PSO variants.

Since exchanging information between different swarms can enhance the swarm diversity, a few multi-swarm based PSO have been proposed. For instance, in Liang and Suganthan (2005), the authors tried to dynamically re-group the swarm into multiple sub-swarms to achieve information exchange. Ye et al. (2017) devised a multi-swarm based PSO, in which the particles in each sub-swarm are further classified into ordinary and communication particles. In this method, ordinary and communication particles of each sub-swarm are set to focus on exploitation and exploration, respectively. Zhang and Ding (2011) employed four sub-swarms for evolution, which exchange information to maintain cooperation and to guide their own evolution. Niu et al. (2007) presented a PSO with master-slave model of swarms, which consists of one master swarm and several slave swarms. The master swarm updates the states of particles based on both its own experience and that of the most successful particles in the slave swarms.

2.2 Evolutionary algorithms for LSGO

Traditional evolutionary algorithms (EAs) are generally suitable to address low-dimensional optimization problems. However, they could perform poorly on high-dimensional optimization problems. To address this issue, many extensions of EAs have been proposed and they can be roughly divided into two categories: decomposition and non-decomposition based methods. Decomposition-based methods are usually based on cooperative coevolutionary (CC) framework (Potter 1997) and referred to as cooperative coevolutionary algorithms (CCEAs). These methods adopt a divide-and-conquer strategy to decompose a high-dimensional problem into multiple low-dimensional problems and solve them separately. The optimal solution of each sub-problem is then merged into a global optimal vector called "context vector". For example, in Bergh and Engelbrecht (2004), the authors devised two CC based PSO algorithms, named CCPSO- S_k and CCPSO- H_k . CCPSO- S_k is a CC-based standard PSO, while CCPSO- H_k is a hybrid of CCPSO- S_k and standard PSO. In these methods, the correlation between variables is considered to divide decision variables into K

groups for optimization. Since the decomposition strategy of variables is vital, many CCEAs mainly focus on the decomposition strategy. In Yang et al. (2008), the authors introduced a random grouping scheme, which randomly divides all variables into K groups at each cycle. To address the setting of appropriate group size K , a multilevel CC framework was further devised by Yang et al. (2008). In this method, at the beginning of each cycle, a decomposer is selected from the pool based on the performance of different decomposers. At the end of each cycle, the performance of selected decomposer will be updated. Li and Yao (2012) adopted a simple approach to deal with the issue of group size K . In this method, if the selected decomposer performs well, then it will be continuously employed. Otherwise, a new one will be randomly selected from the pool. Recently, Omidvar et al. (2014) proposed a grouping strategy, which can uncover the underlying interaction structure of variables and form groups such that the interdependence between them is kept to be a minimum. Similar decomposition strategies can also be found in Omidvar et al. (2014, 2017).

Although CCEAs are promising for LSGO, they are generally expensive especially facing a large number of sub-problems, which has to be optimized individually. To alleviate this issue, non-decomposition based methods have also been developed. These methods generally based on *lBest* particle learning (Liang and Suganthan 2005) or inter-particle learning schemes (Cheng and Jin 2015b; Yang et al. 2016, 2018) to enhance the swarm diversity. In Liang and Suganthan (2005), the authors tried to divide the swarm into multiple sub-swarms and each sub-swarm evolves individually based on a *lBest* learning strategy proposed in Kennedy and Mendes (2002). The resulting scheme is able to enhance the diversity of swarm, thus delivering much better performance than standard PSO, which adopts *pBest* and *gBest* for particle learning. The performance of this method, however, is limited for LSGO. In Cheng and Jin (2015a; b), the authors developed a competitive learning strategy for PSO, resulting in a method called competitive swarm optimizer (CSO). In this method, two particles, which are randomly taken from the swarm, are set to compete. The winner X_w will be passed directly to the swarm for evolution, while the loser X_l is set to learn from X_w , as follow:

$$v_{lj} = r_{1j} \cdot v_{lj} + r_{2j} \cdot (x_{w,j} - x_{lj}) + \varphi \cdot r_{3j} \cdot (\bar{x}_j - x_{lj}), \quad (6)$$

where r_1 , r_2 and r_3 are randomly generated values between $[0, 1]$, φ is a parameter controlling learning rate and \bar{x} denotes the mean position of swarm. In Cheng and Jin (2015a; b), the authors presented a PSO with a social learning strategy (SL-PSO). In this strategy, the particle learning of X_i is performed by randomly selecting a particle X_k , which possesses a higher fitness than X_i , from the swarm and updating its velocity as:

$$v_{lj} = r_{1j} \cdot v_{lj} + r_{2j} \cdot (x_{k,j} - x_{lj}) + \varphi \cdot r_{3j} \cdot (\bar{x}_j - x_{lj}). \quad (7)$$

In Yang et al. (2018), the authors devised two swarm optimizers with level-based learning (LLSO) and segment-based predominant learning (SPLSO), respectively. In LLSO, particles are grouped into various levels based on their fitness and each particle is set to learn from two predominant particles from higher levels. While in SPLSO, variables of particles are partitioned into multiple segments randomly and different segments are set to learn from different predominant particles. The above inter-particle learning strategies can be used to improve swarm diversity and the resulting algorithms are able to outperform CCEAs and *lBest* particle learning based PSO. However, these algorithms do not consider the fact that, during evolution, different particles in the swarm generally have different potentials in searching the space, thus should be treated differently in terms of being selected for evolution and learning. This could limit their performance.

Except for PSO, other EAs and their variants have also been developed for LSGO. For instances, Molina et al. (2010) tried to combine a local search into a steady-state GA, resulting an algorithm termed as MA-SW-Chains, to deal with LSGO. LaTorre et al. (2012) developed a multiple offspring sampling framework, which hybridizes multiple algorithms to deal with LSGO. Maucec et al. (2018) incorporated three evolution strategies into differential evolution for LSGO. Yildiz and Topal (2019) proposed a micro differential evolution algorithm with a directional local search operator using a small population size to solve LSGO. A good review of EAs for LSGO can be found in LaTorre et al. (2015).

3 Proposed method

In this section, we propose an attention-based particle swarm optimizer (APSO) for LSGO. In the proposed algorithm, an attention-based particle sampling (APS) strategy is designed and employed to dynamically activate an appropriate sub-swarm to participate in evolution at each generation. The activated sub-swarm will then go through an attention-based particle learning (APL) strategy. The APL works by randomly selecting three predominant particles from a sub-swarm, which is activated by the attention mechanism, to guide the learning of particles. The evolution will process until the termination condition is met. Algorithm 1 shows the procedure of the proposed algorithm. The details of APS and APL strategies in our algorithm are given in the follow sections.

- Step 1. Generate an initial swarm P with Np solutions.
- Step 2. Select a sub-swarm $subP$ from P using the proposed APS strategy (see Section 3.1):
- i. Sort particles in the swarm in descending order based on their fitness.
 - ii. For each particle in descending order:
 - a) Calculate its activation value according to equation (8).
 - b) Generate a randomly value between 0 to 1. If the calculated activation value is larger than the randomly generated value, then select the particle as a member of $subP$.
- Step 3. For each particle m in $subP$, perform the devised APL strategy (see Section 3.2):
- i. Select a predominant sub-swarm, PS , using the attention mechanism.
 - ii. For each dimension of the problem, randomly select three predominant particles from PS .
 - iii. Sort the three particles in descending order based on their fitness and label them as e_1 , e_2 and e_3 , respectively.
 - iv. Perform equations (9) and (10) to update the velocity and position, respectively, of m .
- Step 4. Repeat steps 2 to 4 until the termination condition is met.
- Step 5. Output the best solution in P .

Algorithm 1. An attention-based particle swarm optimizer for LSGO.

3.1 Attention-based particle sampling

To tackle LSGO, a swarm optimizer is required to preserve a sufficient swarm diversity during evolution, so that the solution space could be well explored to avoid local optima. At the same time, the optimizer should efficiently exploit promising areas of solution space to locate the optimal or near optimal solution of the problem. These two requirements generally conflict with each other (Cheng and Jin 2015a; Campos et al. 2014) and a good optimizer should have a balanced exploration and exploitation capability to appropriate search the solution space.

To support a well-balanced PSO evolution, here we first devise an attention-based particle sampling strategy, which is inspired by the attention mechanism. In cognitive science, to make rational usage of limited visual information processing resources, humans usually selectively focus on a portion of information while ignoring other information to achieve an efficient information process (Itti et al. 1998; Corbetta and Shulman 2002). Similarly, during evolution of PSO, different particles are usually in different evolution states and have different potentials in exploring and exploiting the search space. Paying different attentions to different particles can lead the swarm to different levels of exploration and exploitation during evolution. Motivated by the above mechanism and rationale, an APS strategy, which tends to adaptively activate an appropriate subswarm at different stage of evolution such that encouraging exploration at the early stage of evolution while exploitation at the later stage of evolution, has been developed and incorporated into the proposed method.

Specifically, the proposed APS strategy works as follows. At each generation of evolution, we first sort the particles of

swarm in descending order according to their fitness values. The sorted particles are then ranked from 1 to Np . Subsequently, for each particle i , we calculate its activation value p_i as:

$$p_i = p_{i,ini} + (p_{i,fin} - p_{i,ini}) \times (FEs/FE_{max}), \quad (8)$$

where FE_{max} denotes the maximum number of function evaluations and FEs represents the number of function evaluations consumed so far during evolution. Here, the initial activation value $p_{i,ini}$ is defined as $p_{i,ini} = rank(i)/Np$ while the final activation value $p_{i,fin}$ is computed to be $p_{i,fin} = 1 - p_{i,ini}$. Based on the calculated activation value of each particle, a randomly value between 0 to 1 is then generated. If the activation value of the particle is larger than the randomly generated value, it will be sampled as a member of sub-swarm for evolution. The procedure of the APS is shown in step 2 of Algorithm 1.

It can be found that, according to the above procedure, the particles with lower ranks (i.e., lower fitness) will have higher possibilities to be activated at the early phase of evolution. As the particles with lower fitness are generally scatter around the space and far away from the potential optima, by activating these particles for evolution, the search is thus encouraged to explore the space to locate promising areas. While, during the later stage of evolution, the particles with higher ranks (i.e., higher fitness) will have higher possibilities to be activated. Since the particles with higher fitness are generally locate near potential optima, by activating these particles for evolution, the search is therefore encouraged to exploit the space to accurately identify the optima. Consequently, the proposed APS can be used to support a balanced evolutionary search at swarm level.

3.2 Attention-based particle learning

After obtaining a subset swarm, $supP$, by employing APS at each generation, each particle in $supP$ will go through an attention-based particle learning, which works as follows. For each particle m in $supP$, a predominant sub-swarm, PS , is first selected based on the attention mechanism from the swarm. Specifically, for each particle in the swarm, which is better than the particle m , a randomly value between 0 to 1 is generated. If the randomly generated value is less than the particle's activation value, then insert the particle into PS . After obtaining the predominant sub-swarm, three exemplars will be randomly selected from PS and compete to update the particle m on each dimension of the problem. The competition is performed by determining the best, medium, and worst particle, denoted as e_1 , e_2 , and e_3 , respectively. Based on the three selected exemplars e_1 , e_2 , and e_3 , the particle m on the j th dimension of the problem will be updated according to following rules:

$$v_{m,j} = r_{1,j} \cdot v_{m,j} + r_{2,j} \cdot (x_{e_1,j} - x_{m,j}) + \varphi_1 \cdot r_{3,j} \cdot (x_{e_2,j} - x_{m,j}) + \varphi_2 \cdot r_{4,j} \cdot (x_{e_3,j} - x_{m,j}), \quad (9)$$

$$x_{m,j} = x_{m,j} + v_{m,j}, \quad (10)$$

where φ_1 and φ_2 denote learning rates, and j denotes the j th dimension of the problem. It should be noted that, the three predominant exemplars will be independently selected from PS on each dimension of the problem for particle learning. By doing so, the diversity of swarm can be further preserved. In case that the size of predominant subswarm PS is less than three, which is required to perform the particle learning, then no particle learning will be carried out. The procedure of the APL is shown in step 3 of Algorithm 1.

Based on the above procedure, for each particle subjected to learning, exemplars are set to be selected from a predominant sub-swarm PS sampled by the attention mechanism. According to the attention mechanism, at the early phase of evolution, the particles with worse fitness will have higher possibilities to be chosen as candidate exemplars. While, at the later phase of evolution, the particles with better fitness will have higher possibilities to be selected as members of PS . Consequently, exemplars selected for particle learning at the beginning of evolution generally have low fitness, thus encouraging the exploration aspect of evolution. While, at the later stage of evolution, exemplars selected for particle learning typically have high fitness, thus advocating exploitation. The APL can therefore be used to achieve a balanced evolutionary search at particle level.

Table 1 Comparing results of APSO and its two variants on 1000-D CEC'2010 functions in term of mean fitness value

Function	APSO	APSO_1	APSO_2
F1	1.45e-01	2.20e-17	1.67e-1
F2	8.60e+02	5.05e+03	6.49e+03
F3	4.02e-14	2.89e-01	2.76e+00
F4	1.15e+11	3.02e+11	4.31e+11
F5	1.56e+07	2.83e+08	3.85e+08
F6	2.01e+01	1.55e+00	1.53e+03
F7	3.02e+04	1.45e+03	8.50e+04
F8	2.08e+05	2.31e+07	4.68e+07
F9	1.26e+07	4.41e+07	2.12e+08
F10	9.12e+02	1.03e+04	6.24e+03
F11	2.05e+01	2.39e+00	9.71e+01
F12	8.00e+02	4.70e+04	2.64e+04
F13	3.48e+02	8.06e+02	2.70e+03
F14	3.70e+07	1.31e+08	5.82e+08
F15	1.06e+03	1.08e+04	2.06e+04
F16	6.67e+00	6.38e+00	4.29e+01
F17	2.83e+04	2.41e+05	3.39e+05
F18	1.04e+03	2.48e+03	4.36e+03
F19	2.25e+06	1.25e+07	5.97e+06
F20	9.34e+02	1.40e+03	6.92e+03

Table 2 Comparing results of APSO and its two variants on 1000-D CEC'2013 functions in terms of mean fitness value

Function	APSO	APSO_1	APSO_2
F1	1.07e-21	2.40e-17	2.28e-16
F2	1.12e+03	3.07e+03	4.31e+03
F3	2.08e+01	2.16e+01	2.23e+01
F4	9.33e+08	7.68e+09	1.21e+10
F5	6.90e+05	6.00e+05	8.64e+05
F6	1.04e+06	1.06e+06	1.06e+06
F7	7.09e+05	7.97e+06	9.57e+06
F8	2.64e+13	8.67e+13	2.12e+14
F9	5.96e+07	6.73e+08	7.34e+08
F10	9.06e+07	9.40e+07	9.53e+07
F11	9.59e+11	9.30e+11	9.33e+11
F12	9.79e+02	1.32e+03	3.64e+03
F13	2.17e+07	2.33e+09	1.98e+09
F14	2.89e+07	1.42e+09	2.35e+09
F15	6.11e+06	1.50e+08	1.56e+08

4 Experiments

In this section, we carry out a series of experiments to evaluate the proposed method and compare it with related

Table 3 Comparing results of various algorithms on 1000-D CEC'2010 functions

Function	Index	APSO	SL-PSO	CSO	DSPLSO	DLLSO	MA-SW-Chains	DECC-DG	MLCC	DECC-G	CCPSO2
F1	Mean	1.45e-01	8.73e-18	4.50e-12	7.73e-20	3.13e-22	9.75e-20	1.88e+04	8.65e-13	3.54e-07	2.96e+00
	Std	6.28e-01	3.30e-18	5.94e-13	7.07e-21	8.03e-23	3.33e-19	4.66e+04	2.97e-12	1.44e-07	6.68e+00
	<i>t</i> -test	-	-1.26e+00	-1.26e+00	-1.26e+00	-1.26e+00	-1.26e+00	-1.26e+00	2.21e+00	-1.26e+00	-1.26e+00
F2	Mean	8.60e+02	1.93e+03	7.42e+03	4.45e+02	9.82e+02	5.74e+02	4.43e+03	2.89e+00	1.33e+03	4.30e+00
	Std	3.84e+01	1.12e+02	2.86e+02	1.65e+01	4.39e+01	1.41e+02	1.87e+02	1.52e+00	2.55e+01	1.11e+00
	<i>t</i> -test	-	4.95e+01	1.25e+02	-5.44e+01	1.15e+01	-1.07e+01	1.02e+02	-1.22e+02	5.58e+01	-1.22e+02
F3	Mean	4.02e-14	1.88e+00	2.60e-09	2.52e-13	2.76e-14	1.12e-12	1.66e+01	2.10e-07	1.10e+00	4.51e-03
	Std	5.14e-15	3.30e-01	2.62e-10	1.89e-14	2.38e-15	5.73e-13	3.02e-01	1.12e-06	3.35e-01	1.66e-03
	<i>t</i> -test	-	3.12e+01	5.44e+01	5.92e+01	-1.22e+01	1.03e+01	3.01e+02	1.03e+00	1.80e+01	1.49e+01
F4	Mean	1.15e+11	2.99e+11	7.25e+11	4.30e+11	4.40e+11	2.74e+11	5.22e+12	1.71e+13	2.59e+13	1.70e+12
	Std	2.99e+10	7.16e+10	1.23e+11	8.31e+10	1.10e+11	7.24e+10	1.89e+12	5.47e+12	8.14e+12	1.04e+12
	<i>t</i> -test	-	1.30e+01	2.64e+01	1.95e+01	1.56e+01	1.11e+01	1.48e+01	1.70e+01	1.74e+01	8.34e+00
F5	Mean	1.56e+07	3.17e+07	2.86e+06	6.30e+06	1.22e+07	3.42e+07	1.55e+08	4.99e+08	2.69e+08	4.14e+08
	Std	4.41e+06	6.21e+06	1.79e+06	1.76e+06	3.43e+06	6.69e+06	2.15e+07	1.07e+08	6.84e+07	1.38e+08
	<i>t</i> -test	-	1.16e+01	-1.47e+01	-1.07e+01	-3.33e+00	-3.33e+00	1.27e+01	2.47e+01	2.02e+01	1.58e+01
F6	Mean	2.01e+01	2.08e+01	8.21e-07	9.45e-09	5.20e-01	1.41e+05	1.63e+01	1.78e+07	5.00e+06	1.71e+07
	Std	2.55e-02	2.63e+00	2.68e-08	1.20e-09	7.46e-01	3.67e+05	3.45e-01	4.37e+06	1.03e+06	5.20e+06
	<i>t</i> -test	-	1.46e+00	-4.32e+03	-4.32e+03	-1.44e+02	-1.44e+02	2.10e+00	2.23e+01	2.66e+01	1.80e+01
F7	Mean	3.02e+04	6.49e+04	2.01e+04	4.76e+02	7.19e+02	1.04e+01	1.41e+04	1.51e+08	8.14e+08	2.06e+08
	Std	3.38e+04	5.60e+04	3.86e+03	1.31e+02	2.59e+03	4.02e+00	1.26e+04	1.45e+08	5.40e+08	4.31e+08
	<i>t</i> -test	-	2.91e+00	-1.63e+00	-4.82e+00	-4.76e+00	-4.89e+00	-2.44e+00	5.70e+00	8.26e+00	2.62e+00
F8	Mean	2.08e+05	7.81e+06	3.87e+07	3.11e+07	2.34e+07	1.15e+07	2.75e+07	6.59e+07	8.56e+07	4.13e+07
	Std	9.54e+04	1.56e+06	6.81e+04	9.36e+04	2.46e+05	2.04e+07	2.63e+07	3.40e+07	2.64e+07	3.84e+07
	<i>t</i> -test	-	2.66e+01	1.80e+03	1.27e+03	4.81e+02	3.03e+00	5.68e+00	1.06e+01	1.77e+01	5.86e+00
F9	Mean	1.26e+07	3.30e+07	7.03e+07	4.59e+07	4.36e+07	3.07e+07	5.59e+07	2.43e+08	4.40e+08	1.02e+08
	Std	1.46e+06	4.46e+06	5.73e+06	3.04e+06	4.28e+06	3.19e+06	6.45e+06	2.16e+07	4.87e+07	3.30e+07
	<i>t</i> -test	-	2.38e+01	5.34e+01	5.41e+01	3.75e+01	2.83e+01	3.59e+01	5.83e+01	4.80e+01	1.48e+01
F10	Mean	9.12e+02	2.56e+03	9.60e+03	7.99e+03	8.91e+02	1.33e+03	4.49e+03	4.24e+03	1.03e+04	5.09e+03
	Std	4.01e+01	2.17e+02	7.67e+01	1.28e+02	3.66e+01	5.67e+01	1.29e+02	1.45e+03	3.13e+02	7.81e+02
	<i>t</i> -test	-	4.09e+01	5.50e+02	2.89e+02	-2.12e+00	-2.12e+00	1.45e+02	1.26e+01	1.63e+02	2.93e+01
F11	Mean	2.05e+01	2.32e+01	4.02e-08	3.04e-12	5.80e+00	8.66e+00	1.02e+01	1.98e+02	2.59e+01	1.98e+02
	Std	1.36e+00	2.10e+00	5.12e-09	2.89e-13	5.40e+00	3.25e+00	8.71e-01	1.12e+00	1.73e+00	2.12e+00
	<i>t</i> -test	-	5.91e+00	-8.26e+01	-8.26e+01	-1.45e+01	-1.84e+01	-3.49e+01	5.52e+02	1.34e+01	3.86e+02
F12	Mean	8.00e+02	1.75e+04	4.37e+05	9.52e+04	1.25e+04	6.34e+04	2.84e+03	1.03e+05	9.55e+04	3.39e+04
	Std	2.01e+02	9.07e+03	6.22e+04	6.69e+03	1.46e+03	1.00e+04	1.08e+03	1.57e+04	9.55e+03	1.19e+04
	<i>t</i> -test	-	1.01e+01	3.84e+01	7.73e+01	4.35e+01	3.43e+01	1.02e+01	3.57e+01	5.43e+01	1.52e+01

Table 4 Comparing results of various algorithms on 1000-D CEC'2013 functions

Function	Index	APSO	SL-PSO	CSO	DSPLSO	DLLSO	MA-SW-Chains	DECC-DG	MLCC	DECC-G	CCPSO2
F1	Mean	1.07e-21	1.09e-17	7.71e-12	1.18e-19	3.99e-22	1.19e-20	6.42e+03	8.60e-10	3.14e-06	4.11e+01
	Std	1.29e-22	2.50e-18	1.31e-12	1.06e-20	1.32e-22	1.11e-20	1.81e+04	4.38e-09	4.27e-06	3.14e+01
	<i>t</i> -test	-	2.39e+01	3.22e+01	6.04e+01	-1.99e+01	5.34e+00	1.94e+00	1.08e+00	4.03e+00	7.17e+00
F2	Mean	1.12e+03	2.13e+03	8.55e+03	1.06e+03	1.14e+03	6.97e+02	1.27e+04	3.82e+00	1.31e+03	3.50e+01
	Std	8.76e+01	1.36e+02	2.65e+02	4.45e+02	5.78e+01	5.51e+01	7.20e+02	1.73e+00	3.63e+01	4.85e+00
	<i>t</i> -test	-	3.42e+01	1.46e+02	-7.25e-01	1.04e+00	-2.24e+01	8.74e+01	-6.98e+01	1.10e+01	-6.77e+01
F3	Mean	2.08e+01	2.16e+01	2.16e+01	2.16e+01	2.16e+01	2.03e+01	2.14e+01	2.00e+01	2.02e+01	2.00e+01
	Std	3.69e-02	1.45e-02	6.15e-03	7.53e-03	4.07e-03	4.36e-02	1.45e-02	2.76e-04	6.18e-03	1.25e-04
	<i>t</i> -test	-	1.11e+02	1.17e+02	1.16e+02	1.18e+02	-4.79e+01	8.29e+01	-1.19e+02	-8.78e+01	-1.19e+02
F4	Mean	9.33e+08	4.35e+09	1.32e+10	9.40e+09	6.68e+09	5.13e+09	7.70e+10	2.34e+11	2.35e+11	3.49e+10
	Std	3.10e+08	9.48e+08	2.54e+09	1.89e+09	1.68e+09	1.33e+09	2.82e+10	1.26e+11	1.22e+11	2.17e+10
	<i>t</i> -test	-	1.88e+01	2.63e+01	2.42e+01	1.84e+01	1.68e+01	1.48e+01	1.01e+01	1.05e+01	8.57e+00
F5	Mean	6.90e+05	8.41e+05	5.91e+05	6.30e+05	7.00e+05	1.76e+06	5.78e+06	1.27e+07	8.26e+06	1.40e+07
	Std	1.05e+05	1.75e+05	1.07e+05	1.02e+05	1.28e+05	3.26e+05	3.83e+05	3.46e+06	1.14e+06	4.81e+06
	<i>t</i> -test	-	4.05e+00	-3.62e+00	-2.24e+00	3.31e-01	1.71e+01	7.02e+01	1.90e+01	3.62e+01	1.52e+01
F6	Mean	1.04e+06	1.06e+06	1.06e+06	1.06e+06	1.06e+06	1.05e+06	1.06e+06	1.05e+06	1.06e+06	1.05e+06
	Std	2.97e+04	1.48e+03	1.10e+03	8.05e+02	8.28e+02	7.00e+03	1.07e+03	4.13e+03	1.84e+03	5.24e+03
	<i>t</i> -test	-	3.68e+00	3.69e+00	3.69e+00	3.69e+00	1.80e+00	3.69e+00	1.83e+00	3.68e+00	1.82e+00
F7	Mean	7.09e+05	1.63e+06	5.88e+06	5.50e+06	1.60e+06	2.91e+06	4.78e+08	1.43e+09	1.04e+09	4.15e+08
	Std	2.56e+05	7.05e+05	2.58e+06	2.26e+06	8.38e+05	1.30e+06	1.92e+08	1.07e+09	4.48e+08	9.38e+08
	<i>t</i> -test	-	6.73e+00	1.09e+01	1.15e+01	5.57e+00	9.10e+00	1.36e+01	7.32e+00	1.27e+01	2.42e+00
F8	Mean	2.64e+13	1.03e+14	2.60e+14	1.55e+14	1.20e+14	1.28e+14	3.57e+15	9.59e+15	7.50e+15	1.18e+15
	Std	1.26e+13	3.62e+13	5.87e+13	2.96e+13	3.35e+13	3.44e+13	1.85e+15	6.18e+15	3.18e+15	9.99e+14
	<i>t</i> -test	-	1.09e+01	2.13e+01	2.19e+01	1.43e+01	1.43e+01	1.05e+01	8.48e+00	1.29e+01	6.32e+00
F9	Mean	5.96e+07	8.25e+07	6.06e+07	8.07e+07	1.30e+08	1.09e+08	4.90e+08	9.55e+08	5.96e+08	3.76e+09
	Std	1.19e+07	2.03e+07	1.60e+07	2.24e+07	3.97e+07	1.96e+07	3.18e+07	2.92e+08	9.76e+07	1.02e+09
	<i>t</i> -test	-	5.33e+00	2.75e-01	4.56e+00	9.30e+00	1.18e+01	6.94e+01	1.68e+01	2.99e+01	1.99e+01
F10	Mean	9.06e+07	9.25e+07	9.40e+07	9.39e+07	9.40e+07	9.34e+07	9.45e+07	9.27e+07	9.29e+07	9.30e+07
	Std	1.49e+04	1.67e+06	1.51e+05	2.26e+05	2.11e+05	3.55e+05	2.46e+05	6.07e+05	6.16e+05	7.01e+05
	<i>t</i> -test	-	6.23e+00	1.23e+02	7.98e+01	8.80e+01	4.32e+01	8.67e+01	1.89e+01	2.04e+01	1.87e+01
F11	Mean	9.59e+11	9.33e+11	9.30e+11	9.27e+11	9.30e+11	9.59e+08	4.83e+10	2.28e+11	1.28e+11	9.37e+11
	Std	3.62e+10	1.46e+10	1.03e+10	9.48e+09	9.50e+09	1.68e+09	4.33e+10	1.53e+11	7.15e+10	1.53e+10
	<i>t</i> -test	-	-3.65e+00	-4.22e+00	-4.68e+00	-4.24e+00	-1.45e+02	-8.84e+01	-2.55e+01	-5.68e+01	-3.07e+00
F12	Mean	9.79e+02	1.78e+03	1.07e+03	1.05e+03	1.79e+03	1.33e+03	1.71e+11	2.49e+03	4.35e+03	2.10e+03
	Std	5.27e+01	1.74e+02	7.78e+01	5.37e+01	1.39e+02	1.00e+02	2.24e+10	7.51e+02	7.83e+02	1.78e+02
	<i>t</i> -test	-	2.41e+01	5.30e+00	5.17e+00	2.99e+01	1.70e+01	4.18e+01	1.10e+01	2.35e+01	3.31e+01

Table 3 (continued)

Function	Index	APSO	SL-PSO	CSO	DSPLSO	DLLSO	MA-SW-Chains	DECC-DG	MLCC	DECC-G	CCPSO2
F13	Mean	3.48e+02	9.59e+02	6.29e+02	5.48e+02	7.35e+02	9.89e+02	6.27e+03	4.22e+03	5.96e+03	1.34e+03
	Std	8.78e+01	3.74e+02	2.32e+02	1.69e+02	1.93e+02	4.52e+02	3.65e+03	4.70e+03	4.16e+03	1.72e+02
	t-test	-	8.71e+00	6.20e+00	5.75e+00	1.00e+01	7.62e+00	8.88e+00	4.51e+00	7.39e+00	2.81e+01
F14	Mean	3.70e+07	8.41e+07	2.49e+08	1.60e+08	1.24e+08	1.70e+08	3.42e+08	5.70e+08	9.78e+08	3.06e+08
	Std	2.74e+06	6.31e+06	1.53e+07	8.50e+06	7.38e+06	1.29e+07	2.42e+07	5.50e+07	7.52e+07	1.19e+08
	t-test	-	3.75e+01	7.47e+01	7.54e+01	6.05e+01	5.52e+01	6.86e+01	5.30e+01	6.85e+01	1.24e+01
F15	Mean	1.06e+03	1.12e+04	1.01e+04	9.91e+03	8.33e+02	2.66e+03	5.86e+03	8.90e+03	1.23e+04	1.08e+04
	Std	7.07e+01	8.65e+01	5.23e+01	6.70e+01	4.31e+01	1.50e+02	1.05e+02	2.07e+03	8.24e+02	1.35e+03
	t-test	-	4.97e+02	5.63e+02	4.98e+02	-1.50e+01	5.28e+01	2.08e+02	2.07e+01	7.44e+01	3.95e+01
F16	Mean	6.67e+00	2.51e+01	5.89e-08	4.68e-12	4.25e+00	5.94e+01	7.53e-13	3.81e+02	6.96e+01	3.96e+02
	Std	6.97e+00	1.16e+01	5.61e-09	4.49e-13	2.41e+00	1.37e+01	6.25e-14	5.76e+01	6.43e+00	5.73e-01
	t-test	-	7.46e+00	-5.24e+00	-5.24e+00	-1.80e+00	1.88e+01	-5.24e+00	3.53e+01	3.63e+01	3.05e+02
F17	Mean	2.83e+04	9.00e+04	2.20e+06	6.84e+05	9.05e+04	1.07e+05	4.03e+04	3.49e+05	3.11e+05	1.25e+05
	Std	3.11e+03	1.58e+04	1.55e+05	3.63e+04	3.53e+03	6.75e+04	2.29e+03	3.11e+04	2.24e+04	5.25e+04
	t-test	-	2.10e+01	7.67e+01	9.86e+01	7.24e+01	6.38e+00	1.70e+01	5.62e+01	6.85e+01	1.01e+01
F18	Mean	1.04e+03	2.77e+03	1.73e+03	1.35e+03	2.55e+03	2.53e+03	1.47e+10	1.81e+04	3.83e+04	3.07e+03
	Std	2.52e+02	8.33e+02	5.22e+02	3.87e+02	8.32e+02	6.86e+02	2.03e+09	9.48e+03	1.53e+04	2.45e+02
	t-test	-	1.09e+01	6.52e+00	3.68e+00	9.51e+00	1.12e+01	3.97e+01	9.85e+00	1.33e+01	3.16e+01
F19	Mean	2.25e+06	5.10e+06	1.01e+07	8.20e+06	1.80e+06	5.46e+05	1.75e+06	2.04e+06	1.14e+06	1.52e+06
	Std	1.14e+05	7.05e+05	5.64e+05	4.69e+05	9.96e+04	2.80e+04	1.10e+05	1.42e+05	6.23e+04	7.10e+04
	t-test	-	2.19e+01	7.47e+01	6.75e+01	-1.63e+01	-7.95e+01	-1.73e+01	-6.32e+00	-4.68e+01	-2.98e+01
F20	Mean	9.34e+02	1.85e+03	1.05e+03	1.06e+03	1.88e+03	1.39e+03	6.41e+10	2.30e+03	4.58e+03	2.11e+03
	Std	7.06e+01	2.59e+02	1.49e+02	1.79e+02	1.90e+02	1.47e+02	6.97e+09	2.26e+02	8.25e+02	1.79e+02
	t-test	-	1.87e+01	3.85e+00	3.59e+00	2.56e+01	1.53e+01	5.04e+01	3.16e+01	2.41e+01	3.35e+01
w/t/l	-	18/2/0	14/2/4	13/1/6	10/2/8	15/1/4	15/0/5	16/2/2	18/1/1	18/0/2	

The mean fitness value and standard deviation along with two tailed t-test at a significance level of $\alpha=0.05$ are reported. The bolded t values mean that APSO is significantly better than the corresponding algorithm

Table 4 (continued)

Function	Index	APSO	SL-PSO	CSO	DSPLSO	DLLSO	MA-SW-Chains	DECC-DG	MLCC	DECC-G	CCPSO2
F13	Mean	2.17e+07	4.65e+08	6.67e+08	1.20e+09	3.35e+08	1.04e+09	2.05e+10	1.06e+10	9.35e+09	4.02e+09
	Std	1.45e+07	2.35e+08	2.45e+08	4.99e+08	1.71e+08	3.28e+08	5.53e+09	3.73e+09	2.78e+09	2.31e+09
	t-test	-	1.03e+01	1.44e+01	1.29e+01	1.00e+01	1.70e+01	2.03e+01	1.55e+01	1.84e+01	9.48e+00
F14	Mean	2.89e+07	3.28e+08	3.62e+09	8.31e+09	1.72e+08	6.53e+09	1.92e+10	2.21e+11	1.42e+11	9.10e+10
	Std	1.76e+07	5.17e+08	1.44e+09	6.67e+09	1.38e+08	5.70e+09	1.44e+10	8.54e+10	5.86e+10	8.53e+10
	t-test	-	3.17e+00	1.37e+01	6.80e+00	5.63e+00	6.25e+00	7.29e+00	1.42e+01	1.33e+01	5.84e+00
F15	Mean	6.11e+06	5.86e+07	7.87e+07	4.13e+07	4.48e+06	8.48e+06	9.90e+06	1.61e+07	1.16e+07	4.75e+06
	Std	4.12e+05	6.11e+06	6.50e+06	3.11e+06	3.32e+05	2.25e+06	2.30e+06	1.90e+06	1.26e+06	5.07e+06
	t-test	-	4.69e+01	6.10e+01	6.14e+01	-1.69e+01	5.67e+00	8.88e+00	2.81e+01	2.27e+01	-1.46e+00
w/t/l	-	14/0/1	12/1/2	12/1/2	10/2/3	10/2/3	11/1/3	13/1/1	10/2/3	13/0/2	10/2/3

The mean fitness value and standard deviation along with two tailed t-test at a significance level of $\alpha=0.05$ are reported. The bolded t values mean that APSO is significantly better than the corresponding algorithm

algorithms. All algorithms are coded using C++ and tested on a workstation with an Intel (R) Core™ i7-3630QM CPU at 2.40 GHz running Windows™ 7 operation system. Unless otherwise stated, the results are averaged over 30 independent runs of the algorithms.

4.1 Benchmark functions and parameter settings

Before presenting the experimental results, we first describe benchmark functions used in experiments as well as parameter configurations of the proposed algorithm. The benchmark function sets are from CEC'2010 (Tang et al. 2010) and CEC'2013 (Li et al. 2013), which contain 20 and 15 functions, respectively. The characteristics of these functions can be found in (Tang et al. 2010; Li et al. 2013). The experiments are carried out on the functions with dimension D of 1000. The value of FE_{max} , which is used as the termination condition, is set to be $3000 \times D$. The above configurations are consistent with previous studies on these two benchmark sets, so their results can be cited and fairly compared with our method. The swarm size of our method and its variants is set to be $2 \times (100 + D/10.0)$, due to the employment of APS strategy. The control parameters φ_1 and φ_2 in APS are configured as 0.4 and 0.1, respectively.

4.2 Exploring the proposed method

Firstly, the significance of APS and APL strategies in proposed algorithm is examined. To verify the usefulness of APL, we compared our proposed algorithm, APSO, with its variants: APSO without APL (denoted as APSO_1) and APSO without APS and APL (denoted as APSO_2). In both variants, the social learning strategy, which is from SL-PSO (Cheng and Jin 2015b), is used for particle learning. Tables 1 and 2 show the results of the three algorithms on CEC'2010 and CEC's 2013 functions, respectively.

By comparing the results of APSO and APSO_1, we can see that the APL is able to greatly enhance the performance of the proposed algorithm. By incorporating APL, APSO can locate better solutions than APSO_1 on most of the functions, except F1, F6, F7, F11, F16 from CEC'2010 and F5, F11 from CEC'2013. By comparing APSO_1 with APSO_2, it can be found that APS strategy can help improve the performance of APSO_1. For example, APSO_1 can deliver better solutions than APSO_2 on most functions of CEC'2010 and CEC'2013. From the above results, it can be concluded that the both APS and APL strategies can help improve the search performance, thus effectively identifying the potential optima in the solution space.

4.3 Comparisons with related algorithms

Then, we compare the performance of our algorithm with related methods on CEC'2010 and CEC'2013 LSGO function sets. The methods for comparison include: (1) variants of PSO: SL-PSO (Cheng and Jin 2015a), CSO (Cheng and Jin 2015b), DSPLSO (Yang et al. 2016), DLLSO (Yang et al. 2018); (2) CCEAs: DECC-DG (Omidvar et al. 2014), MLCC (Yang et al. 2008a), DECC-G (Yang et al. 2008b), CCPSO2 (Li and Yao 2012) and (3) CEC'2010 LSGO winner: MA-SW-Chains (Molina et al. 2010). These methods have been performed on the two benchmark function sets of the same dimension D (i.e., $D = 1000$) by using the same FE_{max} (i.e., $3000 \times D$) as the termination condition. Their results can therefore be fairly compared with our method.

The results of above algorithms are shown in Tables 3 and 4, respectively, for function sets of CEC'2010 and CEC'2013. Two-tailed t -tests have been conducted at a significance level of $\alpha = 0.05$ and reported in Tables 3 and 4 to statistically justify the different performance between our algorithm and each of the methods. The bolded t -test values in the tables indicate that our algorithm is significantly better than the corresponding method. The number of wins, losses and ties (denoted as $w/l/t$) of comparing our method with its counterpart methods is summarized in the last row of the tables. Based on the results, it can be found that APSO can outperform all the methods to be compared. Comparing to the four PSO variants, out of 35 functions, APSO delivers significantly better solutions than SL-PSO, CSO, DSPLSO and DLLSO on 32, 26, 25 and 20, respectively. By comparing APSO with four CCEAs, similar results can also be found. Specifically, out of 35 functions, APSO can achieve significantly better solutions on 28, 26, 31 and 28 functions than DECC-DG, MLCC, DECC-G and CCPSO2, respectively. Comparing to MA-SW-Chains, which is a winner algorithm of CEC'2010 LSGO competition, APSO can deliver significantly better solutions on 26 functions. Based on the comparison results, it is clear that our proposed algorithm is the best choice among the ten methods to be compared. The superiority of APSO is mainly due to the incorporation of APS and APL strategies, which can be used to achieve a balanced evolution search at swarm and particle level, respectively, thus properly searching the space of LSGO.

5 Conclusions

In this paper, we have reported the implementation of a PSO with attention-based particle sampling and learning for LSGO. In the proposed method, the attention-based particle sampling strategy, which works by paying attention to low quality particles at the early stage of evolution

while gradually switching to high quality particles at the later stage of evolution, is devised to achieve a balanced evolutionary search at swarm level. While, the attention-based particle learning strategy, which utilize particles from the predominant sub-swarm activated by the attention mechanism for learning, is designed and employed to improve the efficiency of particle learning as well as the balance of evolution at particle level, thus appropriately searching the space. The performance of proposed method has been evaluated via a series of experiments. The results show that proposed mechanisms are able to greatly improve PSO for addressing the LSGO and the resulting algorithm could outperform related algorithms.

The proposed method can be extended in a few directions. Firstly, it is desirable to devise other attention-based partition activation schemes to activate an appropriate sub-swarm for evolution. For instance, apart from the fitness of the particles, their distances to the local best particle in a given topology could also be considered for designing the activation function. Secondly, different exemplar selection schemes with various number of exemplars could also be studied to evaluate the impact of attention-based particle learning on the performance of evolution. Additionally, it would be interesting to extend the proposed algorithm to deal with real and complex optimization problems, for instance, data clustering (Sheng et al. 2020; Wang et al. 2021b), state estimation of networks (Hu et al. 2021a; Jia 2021; Zhao et al. 2021; Zou et al. 2021), sensor filtering fusion (Geng et al. 2021; Mao et al. 2021), and fault detection in network systems (Ju et al. 2021; Hu et al. 2021b).

Declarations

Conflict of interest The authors declared no potential conflicts of interest with respect to the research, authorship, and publication of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Andrews PS (2006) An investigation into mutation operators for particle swarm optimization. In: Proceedings of the IEEE congress on evolutionary computation, p 1044–1051

- Angeline PJ (1998) Using selection to improve particle swarm optimization. In: Proceedings of the IEEE congress on evolutionary computation, p 84–89
- Bergh FVD, Engelbrecht AP (2004) A cooperative approach to particle swarm optimization. *IEEE Trans Evol Comput* 8(3):225–239
- Campos M, Krohling RA, Enriquez I (2014) Bare bones particle swarm optimization with scale matrix adaptation. *IEEE Trans Cybern* 44(9):1567–1578
- Chen YP, Peng WC, Jian MC (2007) Particle swarm optimization with recombination and dynamic linkage discovery. *IEEE Trans Syst Man Cybern B Cybern* 37(6):1460–1470
- Chen W, Zhang J, Lin Y, Chen E (2013) Particle swarm optimization with an aging leader and challengers. *IEEE Trans Evol Comput* 17(2):241–258
- Cheng R, Jin Y (2015a) A competitive swarm optimizer for large scale optimization. *IEEE Trans Cybern* 45(2):191–204
- Cheng R, Jin Y (2015b) A social learning particle swarm optimization algorithm for scalable optimization. *Inf Sci* 291(6):43–60
- Corbetta M, Shulman GL (2002) Control of goal-directed and stimulus-driven attention in the brain. *Nat Rev Neurosci* 3(3):215–229
- Eberhart RC, Shi Y (2001) Tracking and optimizing dynamic systems with particle swarms. In: Proceedings of the IEEE congress on evolutionary computation, p 94–97
- Geng H, Liu H, Ma L, Yi X (2021) Multi-sensor filtering fusion meets censored measurements under a constrained network environment: advances, challenges and prospects. *Int J Syst Sci* 52(16):3410–3436
- Gong Y, Zhang J, Chung H, Chen W, Zhan Z, Li Y, Shi Y (2012) An efficient resource allocation scheme using particle swarm optimization. *IEEE Trans Evol Comput* 16(6):801–816
- Hu J, Jia C, Liu H, Yi X, Liu Y (2021a) A survey on state estimation of complex dynamical networks. *Int J Syst Sci* 52(16):3351–3367
- Hu J, Zhang H, Liu H, Yu X (2021b) A survey on sliding mode control for networked control systems. *Int J Syst Sci* 52(6):1129–1147
- Ishibuchi Z, Salam K (2013) A deterministic particle swarm optimization maximum power point tracker for photovoltaic system under partial shading condition. *IEEE Trans Ind Electron* 60:3195–3206
- Itti L, Koch C, Niebur E (1998) A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans Pattern Anal Mach Intell* 20(11):1254–1259
- Ji D, Wang C, Li J, Dong H (2021) A review: data driven-based fault diagnosis and RUL prediction of petroleum machinery and equipment. *Syst Sci Control Eng* 9(1):724–747
- Jia XC (2021) Resource-efficient and secure distributed state estimation over wireless sensor networks: a survey. *Int J Syst Sci* 52(16):3368–3389
- Ju Y, Tian X, Liu H, Ma L (2021) Fault detection of networked dynamical systems: a survey of trends and techniques. *Int J Syst Sci* 52(16):3390–3409
- Juang CF (2004) A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. *IEEE Trans Syst Man Cybern B Cybern* 34(2):997–1006
- Kennedy J (1999) Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. *Proc IEEE Congr Evol Comput* 3:1931–1938
- Kennedy J, Eberhart R (1995) Particle swarm optimization. *Proc IEEE Int Conf Neural Netw* 4:1942–1948
- Kennedy J, Mendes R (2002) Population structure and particle swarm performance. *Proc IEEE Congr Evol Comput* 2:1671–1676
- LaTorre A, Muelas S, Pena JM (2012) Multiple offspring sampling in large scale global optimization. In: Proceedings of the IEEE congress on evolutionary computation, p 1–8
- LaTorre A, Muelas S, Pena JM (2015) A comprehensive comparison of large scale global optimizers. *Inf Sci* 316:517–549
- Li X, Yao X (2012) Cooperatively coevolving particle swarms for large scale optimization. *IEEE Trans Evol Comput* 16(2):210–224
- Li X, Tang K, Omidvar MN, Yang Z, Qin K (2013) Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization. Technical report, Evolutionary Computation and Machine Learning Group, RMIT University, Australia
- Liang JJ, Suganthan PN (2005) Dynamic multi-swarm particle swarm optimizer with local search. In: Proceedings of the IEEE congress on evolutionary computation, p 522–528
- Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans Evol Comput* 10(3):281–295
- Lu P, Song B, Xu L (2021) Human face recognition based on convolutional neural network and augmented dataset. *Syst Sci Control Eng* 9(s2):29–37
- Mao J, Sun Y, Yi X, Liu H, Ding D (2021) Recursive filtering of networked nonlinear systems: a survey. *Int J Syst Sci* 52(6):1110–1128
- Maucec MS, Brest J, Boskovic B, Kacic Z (2018) Improved differential evolution for large-scale black-box optimization. *IEEE Access* 6:29516–29531
- Mendes R, Kennedy J, Neves J (2004) The fully informed particle swarm: simpler. Maybe Better *IEEE Trans Evol Comput* 8(3):204–210
- Molina D, Lozano M, Herrera F (2010) MA-SW-Chains: memetic algorithm based on local search chains for large scale continuous global optimization. In: Proceedings of the IEEE congress on evolutionary computation, p 1–8
- Niu B, Zhu YL, He XX, Wu H (2007) MCPSO: a multi-swarm cooperative particle swarm optimizer. *Appl Math Comput* 185(2):1050–1062
- Omidvar MN, Li X, Mei Y, Yao X (2014) Cooperative co-evolution with differential grouping for large scale optimization. *IEEE Trans Evol Comput* 18(3):378–393
- Omidvar MN, Yang M, Mei Y, Li X, Yao X (2017) DG2: A faster and more accurate differential grouping for largescale black-box optimization. *IEEE Trans Evol Comput* 21(6):929–942
- Potter MA (1997) The design and analysis of a computational model of cooperative coevolution. PhD dissertation, Dept. Comput. Sci., George Mason Univ., Fairfax, VA, USA
- Ratnaweera A, Halgamuge S, Watson H (2004) Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans Evol Comput* 8(3):240–255
- Shelokar P, Siarry P, Jayaraman VK, Kulkarni BD (2007) Particle swarm and ant colony algorithms hybridized for improved continuous optimization. *Appl Math Comput* 188(1):129–142
- Sheng W, Wang X, Wang Z, Li Q, Zheng Y, Chen S (2020) A differential evolution algorithm with adaptive niching and k -means operation for data clustering. *IEEE Trans Cybern*. <https://doi.org/10.1109/TCYB.2020.3035887>
- Shi Y, Eberhart RC (1998) A modified particle swarm optimizer. In: Proceedings of IEEE world congress on computational intelligence, p 69–73
- Shi Y, Eberhart RC (1999) Empirical study of particle swarm optimization. In: Proceedings of the IEEE congress on evolutionary computation, p 1945–1950
- Shi Y, Eberhart RC (2001) Fuzzy adaptive particle swarm optimization. In: Proceedings of the IEEE congress on evolutionary computation, p 101–106
- Song B, Miao H, Xu L (2021) Path planning for coal mine robot via improved ant colony optimization algorithm. *Syst Sci Control Eng* 9(1):283–289
- Suganthan PN (1999) Particle swarm optimiser with neighborhood operator. *Proc IEEE Congr Evol Comput* 3:1958–1962

- Tang K, Li X, Suganthan PN, Yang Z, Weise T (2010) Benchmark functions for the CEC 2010 special session and competition on large-scale global optimization. Technical report, Nature Inspired Computation and Applications Laboratory, USTC, China
- Wang X, Wang Z, Sheng M, Li Q, Sheng W (2021a) An adaptive and opposite k -means operation based memetic algorithm for data clustering. *Neurocomputing* 437:131–142
- Wang Y, Zou L, Ma L, Zhao Z, Guo J (2021b) A survey on control for Takagi-Sugeno fuzzy systems subject to engineering-oriented complexities. *Syst Sci Control Eng* 9(1):334–349
- Xu L, Song B, Cao M (2021) An improved particle swarm optimization algorithm with adaptive weighted delay velocity. *Syst Sci Control Eng* 9(1):188–197
- Yang Y, Pedersen JO (1997) A comparative study on feature selection in text categorization. In: *Proceedings of ICML*, p 412–420
- Yang Z, Tang K, Yao X (2008a) Large scale evolutionary optimization using cooperative coevolution. *Inf Sci* 178(15):2986–2999
- Yang Z, Tang K, Yao X (2008b) Multilevel cooperative coevolution for large scale optimization. In: *IEEE conference on evolutionary computation*
- Yang Q, Chen W, Gu T, Zhang H, Deng JD, Li Y, Zhang J (2016) Segment-based predominant learning swarm optimizer for large-scale optimization. *IEEE Trans Cybern* 47(9):2896–2910
- Yang Q, Chen W, Deng JD, Li Y, Gu T, Zhang J (2018) A Level-based learning swarm optimizer for large scale optimization. *IEEE Trans Evol Comput* 22(4):578–594
- Ye W, Feng W, Fan S (2017) A novel multi-swarm particle swarm optimization with dynamic learning strategy. *Appl Soft Comput* 61:832–843
- Yildiz YE, Topal AO (2019) Large scale continuous global optimization based on micro differential evolution with local directional search. *Inf Sci* 477:533–544
- Zhan Z, Zhang J, Li Y, Chung HS (2009) Adaptive particle swarm optimization. *IEEE Trans Syst Man Cybern Part B Cybern* 39(6):1362–1381
- Zhang WJ, Xie XF (2003) DEPSO: hybrid particle swarm with differential evolution operator. In: *Proceedings of IEEE conference on systems, man, and cybernetics*, p 3816–3821
- Zhang JZ, Ding XM (2011) A multi-swarm self-adaptive and cooperative particle swarm optimization. *Eng Appl Artif Intell* 24(6):958–967
- Zhao Z, Qian W, Xu X (2021) Stability analysis for delayed neural networks based on a generalized free-weighting matrix integral inequality. *Syst Sci Control Eng* 9(s1):6–13
- Zou L, Wang Z, Hu J, Liu Y, Liu X (2021) Communication-protocol-based analysis and synthesis of networked systems: progress, prospects and challenges. *Int J Syst Sci* 52(14):3013–3034

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.