An Expanded Particle Swarm Optimization Based on Multi-Exemplar and Forgetting Ability

Xuewen Xia[a], Ling Gui[a], Guoliang He*[b], Bo Wei[c], Yinglong Zhang[a], Fei Yu[a], Hongrun Wu[a], Zhi-Hui Zhan*[d,e]

[a]*College of Physics and Information Engineering, Minnan Normal University, Zhangzhou 363000, China*
[b]*School of Computer Science, Wuhan University, Wuhan, 430072, China*
[c]*School of Software, East China Jiaotong University, Nanchang, 330013, China*
[d]*School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510006 China*
[e]*State Key Laboratory of Subtropical Building Science, South China University of Technology, Guangzhou, 510006 China*

Abstract

There are two phenomena in human society and biological systems. One is that people prefer to extract knowledge from multiple exemplars to obtain better learning ability. The other one is the forgetting ability that helps the encoding and consolidation of new information by removing unused or unwanted memories. Inspired by these phenomena, this paper transplants the multi-exemplar and forgetting ability to particle swarm optimization (PSO), and proposes an eXpanded PSO, called XPSO. Firstly, XPSO expands the "social-learning" part of each particle from one exemplar to two exemplars, learning from both the locally and the globally best exemplars. Secondly, XPSO assigns different forgetting abilities to different particles, simulating the forgetting phenomenon in the human society. Under the multi-exemplar learning model with forgetting ability, XPSO further adopts an adaptive scheme to update the acceleration coefficients and selects a reselection mechanism to update the population topology. The effectiveness of these new proposed strategies is verified by extensive experiments. Moreover, the comparison results among XPSO and other 9 popular PSO as well as 3 non-PSO algorithms on CEC'13 test suite suggest that XPSO attains a very promising performance for solving different types of functions, contributing to both higher solution accuracy and faster convergence speed.

Keywords: Particle swarm optimization, global optimization, multi-exemplar, forgetting ability, adaptive adjustment

1. Introduction

Particle swarm optimization algorithm (PSO) is a widely known swarm intelligence algorithm proposed by Kennedy and Eberhart in 1995 [10, 16]. In PSO, a population searches for solutions by simulating the swarm behaviors of birds flocking. During the optimization process, each particle characterized by a position vector and a velocity vector adjusts its search trajectory based on the historical experience of itself and its neighbors. The intelligence emerged from the simple interaction among particles puts PSO into more and more extensive applications during the last two decades [41].

In PSO, parameters and learning models play crucial roles since they determine particles' search trajectory. Thus, extensive studies focusing on parameters adjustments and learning models tunes appeared in last years. The common idea of these improvements is utilizing time-varying strategies and/or adaptive adjustments to regulate the parameters and learning models. For example, in the most ubiquitous update rules of w, c_1 and c_2 [35, 37], the parameters are adjusted based on the iteration numbers aiming to meet different search requirements of different evolutionary stages. Furthermore, iteration-based topology adjustments also attain favorable comprehensive performance [30].

However, these improvements only consider the iteration number as a criterion for the adjustments as well as over-dependent on knowledge discovery of each particle under a monotonic parameters setting and/or learning models. Thus, the strategies may cause a lack of intelligence in the population to deal with various complex situations. To bring more intelligence to the population, some PSO variants use fitness-based adjustments for parameters [38] and learning models [19, 44]. The motivation of the improvements is that various tuning methods assigned for different particles enable the particles to play different roles on coping with complex situations.

Inspired by the above mentioned modifications, we propose an eXpanded PSO (XPSO) in which three strategies are employed to enhance its comprehensive performance on different problems. First, the single exemplar in the canonical PSO increases to multiple exemplars. Second, when updating velocity, each particle adjusts its acceleration coefficients for different parts based on the population's historical experience rather than the iteration number. Based on the adaptive acceleration coefficients, the local version PSO (LPSO) and the global version PSO (GPSO) can be organically embedded rather then a passive combination. Last, inspired by the forgetting phenomenon appeared in people's learning process, we assign different forgetting abilities for different particles. To imitate the effects of familiarity and age, which display crucial roles in human's forgetting, Euclidean distance and evolutionary stage are used to compute each particle's forgetting ability.

The rest of this paper is organized as follows. Section 2 presents the framework of the canonical PSO and some PSO variants. The motivations and details of XPSO are described in Section 3. In Section 4, the performance of new introduced strategies are experimental verified. Simulation results are presented in Section 5 for the comparison of XPSO with other 9 PSO variants and 3 non-PSO algorithms. Finally, conclusions are presented in Section 6.

2. Related works

In this section, we briefly describe the canonical PSO and review previous works on PSO. Furthermore, to facilitate the explanation of PSO later, some necessary notations and terminologies are also introduced in this section.

2.1. Canonical PSO

In PSO, the position of a particle is regarded as a potential solution of a problem, and the fly process of the particle can be regarded as a search process. In each generation, the i^{th} particle is associated with two vectors, i.e., a position vector $\mathbf{X}_i = [x_{i,1}, x_{i,2}, ..., x_{i,D}]$ and a velocity vector $\mathbf{V}_i = [v_{i,1}, v_{i,2}, ..., v_{i,D}]$, where D represents the dimensionality of a problem under study. The vector $\mathbf{X}_i$ is regarded as a candidate solution to the problem while the vector $\mathbf{V}_i$ is treated as a search direction and step size of the i^{th} particle. During the evolutionary process, the particle adjusts its flight trajectory based on two vectors, i.e., its personal historical best position $\mathbf{PB}_i = [pb_{i,1}, pb_{i,2}, ..., pb_{i,D}]$ and its neighbors' best-so-far position $\mathbf{NB}_i = [nb_{i,1}, nb_{i,2}, ..., nb_{i,D}]$, respectively. The update rules of $\mathbf{V}_i$ and $\mathbf{X}_i$ are defined as (1) and (2), respectively.

$$v_{i,j}^{t+1} = w \cdot v_{i,j}^{t} + c_1 \cdot r_{1,j} \cdot (pb_{i,j}^{t} - x_{i,j}^{t}) + c_2 \cdot r_{2,j} \cdot (nb_{i,j}^{t} - x_{i,j}^{t}) \tag{1}$$

$$x_{i,j}^{t+1} = x_{i,j}^{t} + v_{i,j}^{t+1} \tag{2}$$

where w represents an inertia weight determining how much the previous velocity is preserved; c_1 and c_2 are two acceleration coefficients deciding relative learning weights for $\mathbf{PB}_i$ and $\mathbf{NB}_i$, respectively; $r_{1,j}$ and $r_{2,j}$ are two random numbers uniformly distributed in the interval [0, 1]; $x_{i,j}^{t}$ and $v_{i,j}^{t}$ represent the j^{th} dimension values of $\mathbf{X}_i$ and $\mathbf{V}_i$ at generation t, respectively.

Generally, three parts to the right of Eq. (1) are named as "inertia", "self-cognitive", and "social-learning", respectively.

2.2. Variants of PSO

PSO has been popularly studied due to the effectiveness and simplicity of it, and numerous PSO variants have been developed during the last few years. According to the different objects to be dealt with, the majority of the PSO variants can generally be categorized into three cases, i.e., parameter adjustment, learning model adjustment, and hybridization strategy. Although hybriding P-SO with other population-based optimization algorithms [11, 22, 27, 34, 42, 43] also is a popular and useful method, it does not discussed in this paper because XPSO only covers the parameter and learning model adjustments.

2.2.1. Parameters adjustment

Generally, it is believed that a small w facilitates the exploitation while a larger one is beneficial for the exploration. Thus, it seems natural to adopt a time-varying w to offset the contradiction between the exploration and exploitation abilities of PSO. The most ubiquitous update rule of w, introduced by Shi and Eberhart [37] in 1998, is linear decreasing from 0.9 to 0.4 over the optimization process, which is still applied in most PSO algorithms now. Since the search process of PSO is nonlinear and complicated, a linear decreasing w cannot truly reflect the real optimization process. Hence, some nonlinearly-varying strategies [26, 46] are introduced to tune w. Inspired by the idea of time-varying w, there appeared many PSO variants with time-varying c_1 and c_2 [3, 14, 35]. The experimental results manifest that a larger c_1 is beneficial for exploration at the early stage while a larger c_2 is propitious to exploitation at the later evolution stage.

To layout a more satisfactory adjustment for parameters in PSO, Zhan *et al.* [48] proposed an adaptive PSO (APSO), in which w, c_1, and c_2 are relied on an evolutionary state estimation (ESE), which is decided by the distribution of population and the fitness of particles rather than the iteration number. Extensive experiments have demonstrated that self-adaptive methods, such as fitness-based [38], distribution-based [12, 21], and velocity-based [47] strategies, for the parameters adjustment can yield very promising performances.

2.2.2. Learning model adjustment

In PSO, particles' learning model, i.e., the selection of proper learning exemplars and objective dimensions, plays a crucial role since it determines the way of information diffusion within the population. Generally, learning models in the canonical PSO are traditionally classified into GPSO and LPSO, in which a particle selects the best historical experience of the entire population ($\mathbf{GB}$) and its local neighbors' best positions ($\mathbf{LB}_i$) as its learning exemplar, respectively. Moreover, many studies [25, 28] suggest that a sparse neighbor topology is beneficial for complicated multimodal problems, while a dense neighbor topology is conducive to simple unimodal problems. To further improve the performance of PSO, many researchers adopt information of the entire population to give particles more diversified learning models [7, 49].

Although these modifications of the learning model demonstrate favorable performance in varying degrees, it is unrealistic to choose an optimal static learning model for a specific problem beforehand due to that many real applications are black-box problems as well as the optimization process is a dynamic course. Thus, quite a few dynamic learning models have been proposed in recent years [8, 23, 32]. Extensive experiments indicate that the dynamic selection of learning exemplars does efficiently keep the population diversity, which is beneficial for difficult multimodal problems.

To endow a population with a more reliable performance when dealing with different complicated situations, Li *et al.* [19] proposed a self-learning PSO (SLPSO), in which each particle is assigned four different roles based on its local fitness landscape. Accordingly, the different roles representing different learning models enable a particle to independently deal with various situations at different optimization processes.

3. Expanded PSO (XPSO)

3.1. Motivation of XPSO

3.1.1. Multiple exemplars in social learning

In the canonical PSO, each particle only chooses one exemplar in its "social-learning" part. However, it is a more common phenomenon in human society that people always have multiple exemplars during their social learning process. Moreover, many studies indicate that children with multiple learning exemplars can offer better word learning ability compared to those children who are only learning from one exemplar [2, 4, 39]. The simplest explanation is that an individual can extract much more useful knowledge from multiple information providers, i.e., multiple exemplars, than from a single exemplar. In fact, the multi-exemplar model has been applied in many PSO researches. For instance, a particle selects different particles' historical best information as its exemplars in CLPSO [23]. In FIPSO [17], a particle is influenced by all its neighbors in different degrees determined by the performance of the neighbors.

Inspired by the phenomenon in human society and some researches in PSO community, we allow each particle has more than one exemplar in its "social-learning" part. Concretely, considering that LPSO and GPSO have their own distinct advantages, we choose $\mathbf{GB}$ and $\mathbf{LB}_i$ as two exemplars for the i^{th} particle's "social-learning" part in this research.

3.1.2. Forgetting ability

Forgetting is a common biological phenomenon, which has attracted much attention of biologists lasted for many decades [6, 36]. As one part of the brain's memory management system, forgetting provides balance to the encoding and consolidation of new information by removing unused or unwanted memories or by suppressing their expression [36]. Some psychological experiments manifest that people who are good at forgetting unwanted information are good at problem solving [5, 50]. Moreover, a variety of work has further refined the forgetting ability and investigated difference forgetting abilities among different individuals or species [6]. In [1], cognitive neuroscientists conducted a new study inspired by research on child abuse find out that adults who were abused as children by someone they knew are more likely to forget about the abuse than adults who were abused as children by a stranger. In addition, the research also show that forgetting processes continuously degrade along with age.

In recent years, the forgetting ability has been applied in various algorithms to solve different applications [18, 31, 40]. For instance, an iteration-based forgetting factor is applied to make a trade-off between perfect learning and robustness of a trajectory tracking control [40]. In [31], a structure learning with forgetting is proposed to generate skeletal artificial neural networks. The forgetting in this work, which can be regarded as a small magnitude weight, is capable of providing substantial multi-layer feedforward neural network reduction.

The above researches indicate that there are a few basic characteristics of forgetting. The first one is that the level of familiarity plays an important role in forgetting. The second one is that the forgetting ability of an individual continuously degrades along with its age. The last one is the difference of forgetting abilities exists not only between different species, but also between different individuals within a same specie. We believe it is these different learning capabilities caused by various forgetting abilities that increase genetic diversity and thereby are helpful for the evolution of species.

Inspired by the aforementioned biological phenomenon, we also transplants the forgetting ability to PSO. Specifically, when updating velocity, a particle forgets much information from its "social-learning" parts based on its own characteristics. In this way, we wish the population could display more diversified behaviors. Furthermore, the population's historical knowledge is adopted to adjust the acceleration coefficients. The detail of the new proposed PSO is introduced as follows.

3.2. Learning model with forgetting ability

Although there are many strategies for a particle to select proper exemplars, for the purpose of simplicity, we choose $\mathbf{GB}$ and $\mathbf{LB}_i$ as two exemplars for the i^{th} particle's "social-learning" part in XPSO. In addition, each particle has its own acceleration coefficients, which can be regulated during the evolution process. Moreover, we assign a specific forgetting ability $\mathcal{F}_i = [f_{i,1}, f_{i,2}, ..., f_{i,D}]$ for the i^{th} particle. The value of $\mathcal{F}_i$ represents how much information of an exemplar is forgotten by the i^{th} particle. Accordingly, the velocity update of the i^{th} particle is described as (3).

$$\begin{aligned} v_{i,j}^{t+1} = w \cdot v_{i,j}^{t} + cp_i \cdot r_{1,j} \cdot (pb_{i,j}^{t} - x_{i,j}^{t}) + cl_i \cdot r_{2,j} \cdot ((1 - f_{i,j}) \cdot lb_{i,j}^{t} - x_{i,j}^{t}) \\ + cg_i \cdot r_{3,j} \cdot ((1 - f_{i,j}) \cdot gb_{j}^{t} - x_{i,j}^{t}) \end{aligned} \tag{3}$$

where cp_i, cl_i, and cg_i are three acceleration coefficients of the i^{th} particle denoting learning weights for $\mathbf{PB}_i$, $\mathbf{LB}_i$, and $\mathbf{GB}$, respectively; $f_{i,j}$ is a forgetting ability deciding how much information of the j^{th} dimension is forgotten when the i^{th} particle learns from its exemplars $\mathbf{LB}_i$ and $\mathbf{GB}$. In this work, the forgetting ability $\mathcal{F}_i$ determines which dimensions are neglected and how much information in those selected dimensions is lost. Note that *neglected dimension* indicates that the dimension's value is entirely ignored by a learning particle while *information-lost dimension* represents that the dimension's value transmitted to the learning particle is not exactly the same as the original value. The inertia weight w in XPSO linearly decreases from 0.9 to 0.4 during the entire evolutionary process, which is very popular in many PSO variants.

Inspired by the research in [1], the forgetting ability in XPSO depends on the familiarity of two particles, which is measured by the distance between the two particles. However, unlike the forgetting phenomenon described in [1], in which being abused is a negative and harmful experience, the forgetting process in XPSO has an opposite process since learning from other outstanding particles is a positive and helpful experience for a particle. Thus, in XPSO, the greater distance is, the lower degree of familiarity is, and then the forgetting phenomenon is more significant.

When the i^{th} particle performs its "social-learning", the value of $\mathcal{F}_i$ depends on the distance ($dist_i$) between the particle and its exemplars $\mathbf{GB}$. The distance $dist_i$ is defined as (4).

$$dist_i = \sqrt{\sum_{j=1}^{D}(x_{i,j} - gb_j)^2} \tag{4}$$

The greater $dist_i$ is, the more dimensional values of $\mathbf{GB}$ are neglected by the particle, and the more information of the chosen dimensions is lost. For simplicity, we first sort all the particles in ascending order by the values of $dist_i$. Without loss of generality, the sorted population $P = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_N\}$ satisfies the condition $dist_1 \leq dist_2 \leq ... \leq dist_N$. Based on the sorted population, an issue to be dealt with is how many dimensions are neglected when a particle learns information from $\mathbf{GB}$. In this work, the number of neglected dimensions (nd) for the i^{th} particle is defined as (5).

$$nd_i = \begin{cases} 0, & \text{if } i \leq \lfloor N \cdot p \rfloor \\ nd_{min} + \lfloor \frac{(nd_{max} - nd_{min}) \cdot i}{N \cdot (1-p)} \rfloor, & \text{Otherwise} \end{cases} \tag{5}$$

where nd_{max} and nd_{min} are predefined maximum number and minimum number of neglected dimensions, respectively; p is a real number distributed in the interval [0, 1].

From (5) we can see that the first $\lfloor p \cdot N \rfloor$ particles in the sorted population do not ignore any dimensions from their exemplar **GB**. In other words, these particles do not have forgetting ability. On the contrary, the number of neglected dimensions of other $N - \lfloor p \cdot N \rfloor$ particles linearly increases as their orders. For the sake of convenience, the set of the $N - \lfloor p \cdot N \rfloor$ particles is denoted as $\mathcal{S}_p$. If a particle does not belong to $\mathcal{S}_p$, the particle who has no forgetting ability can obtain all information from **GB**. On the contrary, if the particle belongs to $\mathcal{S}_p$, it has a forgetting ability on nd_i dimensions. Without loss of generality, the set of the randomly chosen nd_i dimensions of the i^{th} particle is denoted as $\mathcal{S}_{d_i}$, where $|\mathcal{S}_{d_i}|$ is equal to nd_i.

Particles in the initial evolutionary stage can be regarded as a group of juvenile individuals. In the early stage, the juvenile individuals should have a higher forgetting ability. With gradual mature of the individuals, the faculty of memory of them is getting stronger and stronger. In other words, the forgetting ability of the individuals gradually becomes weaker.

Based on the discussion above, the forgetting ability $\mathcal{F}_i$ is defined as (6).

$$f_{i,j} = \begin{cases} e^{\frac{i}{N}} \cdot (max(\mathbf{X}_j) - min(\mathbf{X}_j)) \cdot 0.05, & \text{if } i \in \mathcal{S}_p \text{ and } j \in \mathcal{S}_{d_i} \\ 0, & \text{Otherwise} \end{cases} \tag{6}$$

where $max(\mathbf{X}_j)$ and $min(\mathbf{X}_j)$ are the maximum and minimum values of the entire population on the j^{th} dimension, respectively.

It can be seen that there are three characteristics in (5) and (6). Firstly, a particle with a larger $dist_i$ (i.e., a lower level of familiarity) in a generation can neglect more dimensions when it learns from **GB** as well as forget much more information on the selected dimensions. Secondly, at the early evolutionary stage, which can be deemed as infancy, the $max(\mathbf{X}_j)$ and $min(\mathbf{X}_j)$ can be very different. In this case, a larger $max(\mathbf{X}_j)$-$min(\mathbf{X}_j)$ is helpful to let a particle have a greater forgetting ability, which is beneficial for exploration. On the contrary, the $max(\mathbf{X}_j)$ and $min(\mathbf{X}_j)$ may be similar, resulting in a smaller $max(\mathbf{X}_j)$-

$min(\mathbf{X}_j)$ value to help the mature particle forget little information during the learning process, which is favorable for exploitation. Thirdly, a same particle can display different forgetting abilities at different evolutionary stages.

3.3. Adaptation of acceleration coefficients

From (3) we can see that there are three acceleration coefficients in XPSO. Furthermore, unlike the canonical PSO in which all individuals have the same acceleration coefficients, each particle in XPSO has its own acceleration coefficients, i.e., cp_i, cl_i, and cg_i, which can be adaptively adjusted based on historical knowledge of $\lfloor p \cdot N \rfloor$ elites. Note that the value of p is the same as that in (5).

When the i^{th} particle adjusting its acceleration coefficients, three real values randomly generated by Gaussian distribution functions $\mathcal{N}(\mu_1, \sigma_1^2)$, $\mathcal{N}(\mu_2, \sigma_2^2)$, and $\mathcal{N}(\mu_3, \sigma_3^2)$ are assigned to cp_i, cl_i, and cg_i, respectively. The values of μ_1, μ_2, and μ_3 are updated according to the summation of corresponding parameters of the elite individuals. The update rule can be defined as (7).

$$\mu_k \leftarrow (1-\eta) \cdot \mu_k + \eta \cdot avg(<\mu_k^{elite}>), 1 \leq k \leq 3 \tag{7}$$

where η denotes a learning weight for knowledge from elite individuals; $<\mu_k^{elite}>$ represents a set of μ_k of the elite particles; $avg(\circ)$ is the average value of $\circ$.

It can be seen from (7) that η determines the degree that elites' experience influences μ_k to the next generation. If η=0, the knowledge of the elites has no affection on the adjustment of μ_k (1$\leq k \leq$3), which may cause some useful historical information of the elites to be ignored. On the contrary, if η=1, the new μ_k only depends on the elites. Thus, a particle may ignore its own experience. The effectiveness of η will be discussed in Section 5.

Considering the sum of two acceleration coefficients as 4.0 is still a popular rule in PSO studies [48], we set the sum of μ_k is approximately equal 4.0. For simplicity, μ_1=μ_2=μ_3=1.35 is adopted at the initial evolutionary stage. Moreover, σ_1, σ_2, and σ_3 use a same constant value in this work for simplicity purposes. Experiments indicate that σ_1=σ_2=σ_3=0.1 can obtain a favorable performance. Due to the space limitation, details of the experimental results are not included in this paper. Thus, the values of cp_i, cl_i, and cg_i can be independently generated by $\mathcal{N}(\mu_1, 0.1^2)$, $\mathcal{N}(\mu_2, 0.1^2)$, and $\mathcal{N}(\mu_3, 0.1^2)$, respectively.

Furthermore, when to adjust the parameters is another crucial issue. The simplest strategy is regulating the parameters in a predefined cycle. However, it is an inappropriate choice that setting a specific cycle for different functions in advance since different problems have their own distinct characteristics. For example, a smaller cycle may waste too much computing resource as well as frequently disturb the promising search model of the population. On the contrary, a greater cycle cannot enable the population to timely regulate its learning strategy when it has ceased improving.

In XPSO, we select the consecutive generations-stagnancy of **GB**, named as $Stag_{GB}$ in this research, as a criteria for the adjustment. Note that, when the performance of **GB** does not improve (i.e., **GB** is stagnant) in a generation, the value of $Stag_{GB}$ is increased by 1. On the contrary, the value of $Stag_{GB}$ is set to 0 if the performance of **GB** improves in a generation. While $Stag_{GB}$ is larger than $Stag_{max}$, we regard that the population has been trapped in a deep local optimum, and it is very hard for the stagnated population to jump out of the local optimum by the old search model. In this condition, particles timely adjusts their acceleration coefficients without having to wait for a predefined cycle are able to change their search direction so as to fly out the local optimum. On the contrary, the particles does not change their parameters if $Stag_{GB}$ is less than $Stag_{max}$ since the current search model remains effective.

3.4. Reselecting neighbors

It is very important for a particle to select proper neighbors since much useful knowledge extracted from the exemplars can provide some constructive guidance for the particle. Inspired by a phenomenon that a gregarious person always has more various knowledge than an unsociable one, many researchers pour attention on dynamic neighbor topology [22, 30, 32], the effectiveness of which has been verified by extensive experiments [25].

While conducting the reselecting neighbors operator, there are two problems need to be deal with. The one issue is when to carry out the operator. In XPSO, we also adopt $Stag_{GB}$ as a criteria to reselect neighbors for particles. In this case, the particles can timely reselect their exemplars while $Stag_{GB}$ is more than $Stag_{max}$, without having to wait for a predefined regrouping period. The other one issue needing to be solved is which particles should be selected as neighbors for a specific particle. For simplicity, we randomly reselect a new LPSO topology for the population. In particular, the random permutated order numbers are assigned to particles, and then a particle selects its left particle and right particle, in terms of the permutated order number, as its neighbors.

3.5. Framework of XPSO

By incorporating the aforementioned components, the pseudo code of XPSO is shown in **Algorithm 1**, and the source code of it can be downloaded from http://github.com/XuewenXia/PSO

There are three new proposed strategies applied to enhance the performance of XPSO. The first is multi-exemplar in "social-learning" part (see Eq. (4)), by which the merits of LPSO and GPSO are organically integrated. The second is forgetting ability (see line 23-24 of Algorithm 1), by which some particles display more favorable exploration capability. The last is adaptation of acceleration coefficients (see line 25-26 of Algorithm 1). In the strategy, much historical information of elites is used to update the acceleration coefficients.

4. Performance of new proposed strategies

In this section, extensive experiments are conducted to analysis the performance of the new introduced parameters and operators. Without loss of generality, minimum problems are adopted in this work.

4.1. Sensitivity of parameters

In this part, we carry out a set of experiments to find out how η, $Stag_{max}$, and p affect the performance of XPSO.

Algorithm 1. XPSO ( )

```
01: /* Initialization */
02: Set $\mu_1$=$\mu_2$=$\mu_3$=1.35, $\sigma$=0.1, $Stag_{GB}$=0, $Stag_{max}$=5, $\eta$=0.2, $p$=0.5;
03: For $i$=1 to $N$ Do
04:    Randomly initialize $\mathbf{V}_i$, $\mathbf{X}_i$, $cp_i$, $cl_i$, and $cg_i$;
05:    Evaluate $f(\mathbf{X}_i)$; $\mathbf{PB}_i = \mathbf{X}_i$;
06: End For
07: Set GB to the current best position of particles;
08: Sort particles according to $dist$;
09: Compute particles' forgetting ability $f_{i,j}$ according to (5) - (6);
10: /* Main Loop */
11: Repeat
12:    For $i$=1 to $N$ Do
13:       Update $\mathbf{V}_i$ and $\mathbf{X}_i$ according to (3) and (2), respectively;
14:       Evaluate $f(\mathbf{X}_i)$; Update $\mathbf{PB}_i$;
15:    End For
16:    Update GB;
17:    If GB is improved Then
18:       $Stag_{GB}$=0;
19:    Else
20:       $Stag_{GB}$=$Stag_{GB}$+1;
21:    End If
22:    If $Stag_{GB} \geq Stag_{max}$ Then
23:       Sort particles according to $dist_i$ $(1 \leq i \leq N)$;
24:       Compute particles' forgetting ability $\mathcal{F}_i$ according to (5) - (6);
25:       Compute $\mu_1$, $\mu_2$, and $\mu_3$ according to $\lfloor p \cdot N \rfloor$ elites;
26:       Update all particles' $cp_i$, $cl_i$, and $cg_i$;
27:       Randomly select a $LPSO$ topology for the population;
28:    End If
29: Until Terminal Condition
```

To separately analyze the effect of each parameter, we specify default values 0.2, 5, and 0.2 for the parameters η, $Stag_{max}$, and p, respectively. When testing the influence of a parameter, other parameters are set as their default values. For instance, to test the effectiveness of p, we use the default values for η (η=0.2) and $Stag_{max}$ ($Stag_{max}$=5). Five benchmark function, including an unimodal functions (*Sphere*), a nonconvex unimodal function (*Rosenbrock*), and 3 multimodal functions (*Ackley*, *Griewank*, and *Rastrigin*) are selected in the experiments. The dimensionality (D) of the 5 test functions, the number of particles (N), and the maximum number of function evaluations (*MaxFEs*) are set to be 30, 50, and 300 000, respectively. The result of each experiment is the average of 30 independent runs.

345 The comparison results for the parameters are demonstrated by Fig. 1.

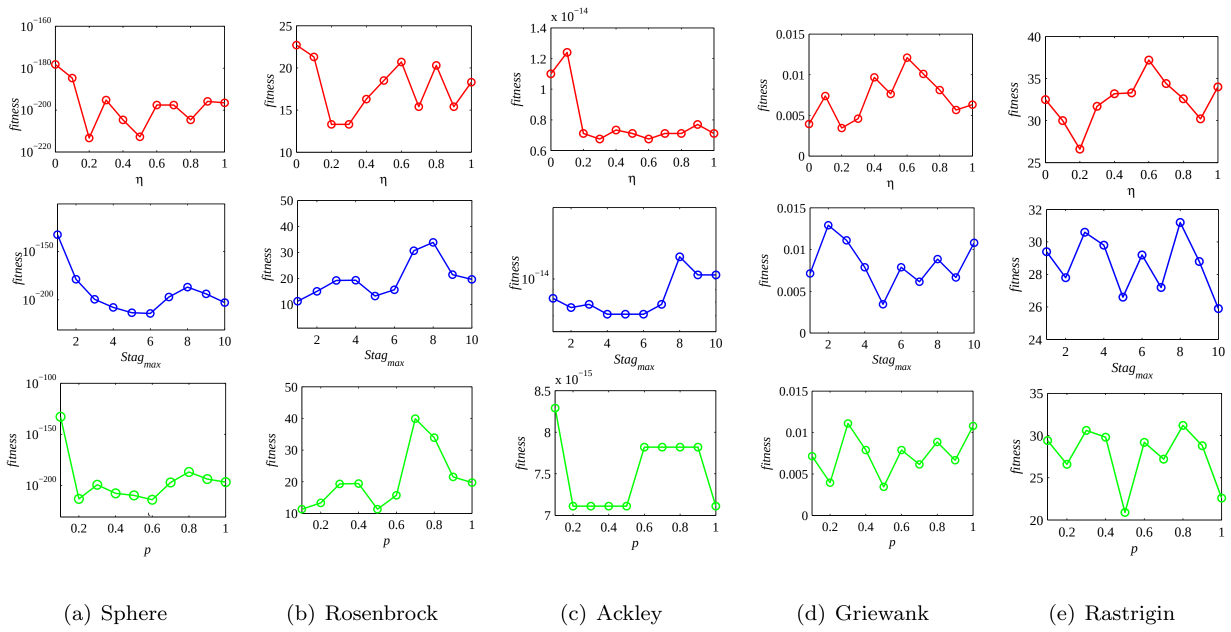


Figure 1: Sensitivity analysis of η, $Stag_{max}$, and p.

4.1.1. Sensitivity of η

The parameter η determines the degree that elites' experience influence a specific particle. In particular, a smaller η causes a particle to learn more information from elites, and then the population has more rapid convergence 350 speed. On the contrary, a greater η enables a particle to extract more knowledge from its own historical experience, which is beneficial for exploration capability. Intuitively, a smaller η should manifest more favorable performance on unimodal functions while a larger one should offer more promising results on multimodal problems. However, the experimental results demonstrated in Fig. 355 1 indicate that a relatively small η has better comprehensive performance on both unimodal and multimodal functions than a larger η. The reason behind the phenomenon may be that over-dependence on elites' knowledge causes particles, especially the leader particle, lose their own promising search direction. It can be observed from the results that setting η in the range of [0.1, 0.3] is beneficial for improving the search efficiency of XPSO on all the test problems except *Griewank*. In this research, η=0.2 is adopted in all the experiments.

4.1.2. Sensitivity of $Stag_{max}$

The value of $Stag_{max}$ determines the frequency of reselecting neighbors for a particle. A smaller $Stag_{max}$ causes the particle to adjust its neighbors more frequently, which may disturb the particle's current search direction. Thus, if the particle has a prospective fly trajectory, the smaller $Stag_{max}$ may cause it to lose its current promising search direction. On the contrary, a greater $Stag_{max}$ is beneficial for maintaining the current search mechanism. However, if the particle has been trapped into a local optimum, the greater $Stag_{max}$ may prevent the particle from changing its learning exemplar, and then jump out of the local optimal solution immediately. The comparison results between different $Stag_{max}$ illustrated in Fig. 1 show that while $Stag_{max}$ equal to 1, which causes a particle to frequently change its neighbors, XPSO has the lowest fitness. Along with the increase of $Stag_{max}$, the performance of it becomes better. However, while $Stag_{max}$ becomes too greater, typically larger than 6, the performance begin to deteriorate. Based on the experiment results, $Stag_{max}$=5 is adopted in this work.

4.1.3. Sensitivity of p

The value of p decides how many elites will provide their knowledge to the population when updating acceleration coefficients. To some extent, p plays a very important role in trade-off between the exploration and the exploitation. A smaller p causes the population to have a more favorable exploitation while a larger p endows the population with a very promising exploration. From the performance curves on *Sphere* demonstrated in Fig. 1 we can see that a relatively small p attains better results. One reason behind this phenomenon may be that there is no local optimal in the unimodal problem, and a small p can help the population rapidly converge to the global optimal solution. On the contrary, for the other 4 problems, neither a smaller p nor a greater p offers the most promising result while setting p in the range of [0.4, 0.5] is beneficial for improving the solution accuracy of XPSO. From the results we discern that $p = 0.5$ is the most superior result.

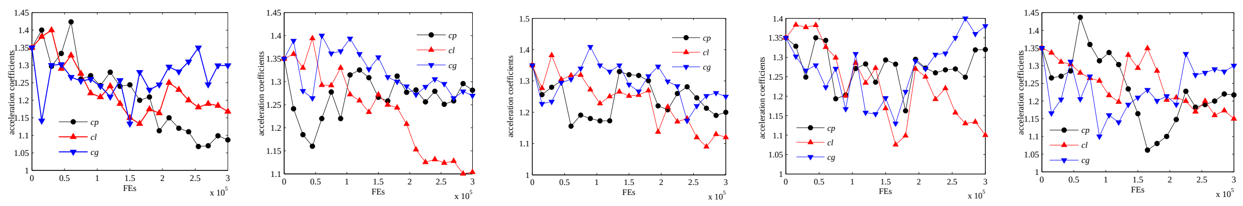
Extensive comparison experiments to verify the comprehensive performance of XPSO is introduced in the following sections, in which we adopt 0.2, 5, and 0.5 for η, $Stag_{max}$, and p, respectively, based on the investigation of the experimental results above.

4.1.4. Change trend of the three acceleration coefficients

In this section, a set of experiments is also conducted to illustrate the change process of the 3 acceleration coefficients on different functions. The results are represented in Fig. 2. Note that the value of each acceleration coefficient is the average result of the top 50% particles measured by fitness.

The results show that the general trend of cl is becoming smaller during the evolutionary process. At the later evolution stage, the value of cl is the smallest one of the three acceleration coefficients. The phenomenon verifies that a relatively great learning weight for **LB** at the initial evolution stage is beneficial for improving the population's exploration. While the population finds out a potential promising region at the later stage, particles pay more attention on exploitation. Thus, the value of cl becomes smaller and smaller. On the contrary, the curve of cg indicates that the learning weight for **GB** becomes greater during the later evolution stage. In particular, cg achieves the greatest value at the last few generations on all the test functions except *Rosenbrock*, on which cg is slightly smaller than cp during the last a few generations. From the variation trend of cp on the 3 multimodal functions we can observe that it oscillates. The reason may be that particles need the oscillated cp to switch between the exploration and the exploitation. On *Sphere*, cp has the smallest value in the last half evolution stage while it has a greater value in the initial stage. It can be explained as that when optimizing a simple unimodal problem, a larger *cp* enables particles to pay more attention on walking around their current position in the initial evolutionary stage. While some elites finding out solutions near the global optimum, other particles pay more attention on extracting useful information from the elites. Thus, *cp* has the smallest value at the end of evolutionary stage.

Note that, although we have a rudimentary understanding of how the different parameters affect the performance of XPSO, the optimal value of each parameter may not be the best parameter setting for a particular problem. Firstly, it is impractical for us to test all possible values of the three parameters. Secondly, when testing the effect of a particular parameter, the set of default values of other parameters may not be the promising one. Thirdly, there may be dependency between the different parameters, which causes the combination of each optimal value no long to be an optimal setting for XPSO. Thus, there is still a lot of space for studying their characteristics, and then provide more suitable regulation strategies.



(a) Sphere (b) Rosenbrock (c) Ackley (d) Griewank (e) Rastrigin

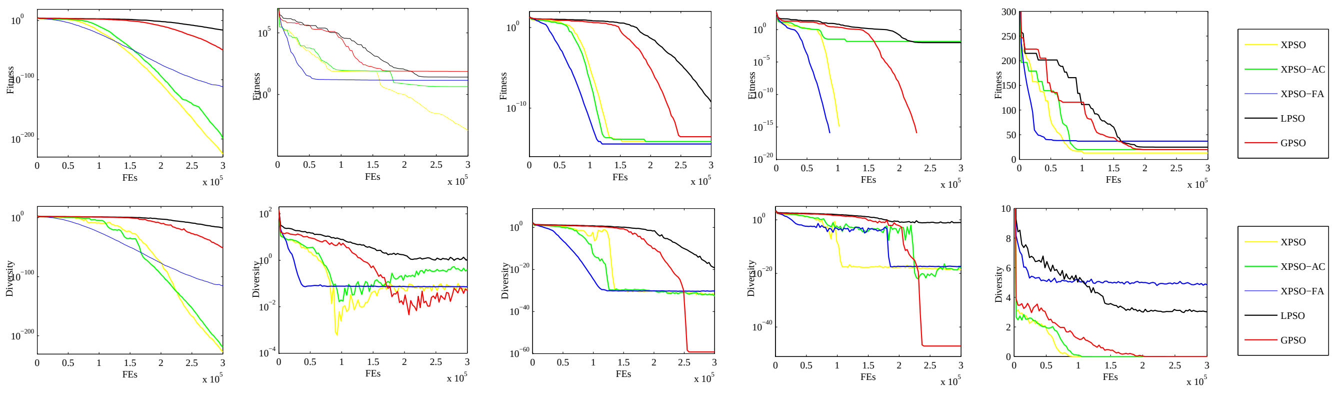
Figure 2: Change process of *cp*, *cl*, and *cg*.

4.2. Effectiveness of operators

There are three operators introduced to improve the performance of XPSO, i.e., multi-exemplar, forgetting ability, and adaptive acceleration coefficients. To find out how the three key strategies affect the performance of XPSO, a set of experiments is conducted on the 5 benchmark functions applied in the last set of experiments. The experimental results are listed in Fig. 3, in which XPSO-FA and XPSO-AC denote algorithms that the forgetting ability and adaptation of acceleration coefficients are removed from XPSO, respectively. Moreover, 440 LPSO and GPSO are also involved in the experiments. Note that the solution accuracy, convergence speed, and the population deviation denoted as (8) are chosen to testify the effectiveness of the new introduced operators.

$$diversity(N) = \frac{1}{N \cdot |L|} \cdot \sum_{i=1}^{N} \sqrt{\sum_{j=1}^{D} (x_{i,j} - \overline{x}_j)^2} \tag{8}$$

where N is the population size, $|L|$ is the length of the longest diagonal in the searching space, D is the dimensionality of the problem, $x_{i,j}$ is the j^{th} value of 445 the i^{th} particle, and $\overline{x}_j$ is the j^{th} value of the average point $\overline{x}$.



(a) Sphere (b) Rosenbrock (c) Ackley (d) Griewank (e) Rastrigin

Figure 3: Comparison results of the new introduced operators.

From Fig. 3 we can see that LPSO shows the best performance on maintaining diversity on all the test functions except *Rastrigin*, on which it is slightly weaker then XPSO-AC. However, LPSO sacrifices performance on solutions accuracy and convergency speed. On the contrary, GPSO loses the diversity more 450 rapidly than other algorithms on 3 functions, especially during the later evolutionary stage. It is worth to note that LPSO and GPSO cannot offer the best results, in terms of solutions accuracy, except GPSO achieves the global optimal solution on *Griewank* with a slower convergence speed. The comparison results among XPSO, XPSO-FA, and XPSO-AC indicate that XPSO offers the best results on 4 out of the 5 functions except *Griewank*, on which it is fared marginally worsen than XPSO-AC measured by solutions accuracy. Moreover, XPSO yields better performance than XPSO-FA on *Sphere*, *Rosenbrock*, *Ackley*, and *Griewank* at the initial evolutionary stage, in terms of diversity. The results verify that the forgetting ability is beneficial for keeping diversity and improving solutions accuracy. In addition, the comparison results between X-PSO and XPSO-AC indicate that they have almost the same performance on keeping diversity on 4 functions. However, XPSO obtains more accurate solutions than XPSO-AC on all the functions except *Griewank*, on which both XPSO and XPSO-AC find out the global optimum solution.

Moreover, to further testify the comprehensive performance of the forgetting ability and the adaptive acceleration coefficients, an extensive experiment is conducted on CEC'13 test suite, in which 28 functions are divided into three groups, including 5 unimodal functions (f_1-f_5), 15 basic multimodal functions (f_6-f_{20}), and 8 composition functions (f_{21}-f_{28}). More detailed information of the functions can refer to the literature [24]. In the experiments, the values of D and *MaxFEs* are 30 and 300000, respectively. Due to the space limitation, only the t-test results between XPSO and the two XPSO variants are presented in Table 1, in which the symbols "(+)", "(-)" and "(=)" denote that XPSO is significantly better than, significantly worse than, and almost the same as the corresponding competitor algorithm, respectively.

The t-test results show that XPSO-FA dominates XPSO on the 4 out of the 5 unimodal functions. On the contrary, XPSO yields more favorable performance than XPSO-FA on the basic multimodal function and the composition functions. The performance between XPSO and XPSO-AC indicates that XPSO displays more comprehensive performance than XPSO-AC on the majority of the test functions.

Table 1: Performance of the forgetting strategy and the adaptive acceleration coefficients

XPSO vs		f_1-f_5	f_6-f_{20}	f_{21}-f_{28}
XPSO-FA	(+)	-	f_7,f_9,f_{10},f_{11},f_{12},f_{17}	f_{24},f_{26},f_{27}
	(=)	f_2	f_6,f_8,f_{14},f_{18},f_{19},f_{20}	f_{21},f_{25},f_{28}
	(-)	f_1,f_3,f_4,f_5	f_{13},f_{15},f_{16}	f_{22},f_{23}
XPSO-AC	(+)	f_1,f_4,f_5	f_7,f_8,f_9,f_{11},f_{12},f_{14},f_{17}	f_{24},f_{26},f_{27},f_{28}
	(=)	f_2	f_6,f_{13},f_{19}	f_{25}
	(-)	f_3	f_{10},f_{15},f_{16},f_{18},f_{20}	f_{21},f_{22},f_{23}

From the experimental results we can obtain some preliminary conclusions for the new introduced operators. Firstly, the multi-exemplar plays a positive role on most of the test functions measured by solution accuracy, convergence speed, and algorithm reliability. Secondly, the forgetting ability is beneficial for maintaining population diversity, though it has a negative performance on solutions accuracy due to that some helpful information of the previous best solution is forgotten. Lastly, the adaptive adjustment of acceleration coefficients contributes to improving solutions accuracy on different types of functions.

5. Experimental studies

5.1. Benchmark functions and peer algorithms

To evaluate different characteristics of XPSO, we carry out different sets of experiments based on CEC'13 test suite. The basic experimental setups in this section are D=30 and $MaxFEs$=10000D.

For comparative analyses, 9 widely accepted PSO algorithms proposed in the last decade are selected as peer algorithms in this study. These peer algorithms are Frankenstein's PSO [30], OLPSO [49], DEPSO [45], PSODDS [15], CCPSO-ISM [20], SRPSO [38], HCLPSO [29], GLPSO [11], and EPSO [27]. The parameters settings and the experimental setup for the peer algorithms are the same as that in the corresponding papers. To describe simply, Frankenstein's PSO is renamed as F_PSO in this study. In the following experiments, the population size of XPSO is N=50. To obtain statistical results, each algorithm is carried out 30 independent runs on each benchmark function.

5.2. Comparison on solutions accuracy

In this section, experiments are conducted to compare the performance of each algorithm in terms of the mean, standard deviations (*Std.Dev*), and median of the fitness. The results are provided in Table 2 - Table 4, where the best results of mean and median values are marked by a shaded background. The mean CPU time usage (*Time*) of each peer algorithm is also reported in the tables.

5.2.1. Unimodal functions (f_1-f_5)

In the first set of experiments, 5 unimodal functions are studied. The comparison results summarized in Table 2 show that XPSO achieves the best results on 2 unimodal functions, i.e., f_3 and f_4, in terms of both mean and median values, followed by OLPSO, PSODDE, and GLPSO. Furthermore, XPSO also yields almost the same as the best solution on f_1 and f_5. From the results we can see that all the algorithms obtain considerable errors on the 3 non-separable problems, i.e., f_2, f_3, and f_4. The complexities lying in the problems are that there are smooth local irregularities and/or smooth but narrow ridges in their fitness landscape besides the correlation between variables. The comparison result manifests that XPSO offers superior performance among all the algorithms on 2 out of the 3 complicated problems. The promising performance of XPSO benefits from the following three advantages. First, comparing with the canonical PSO, XPSO has three exemplars which can give a particle more chance to obtain useful information. Furthermore, the dynamic neighbor topology causes a particle to readjust its search direction endowing the population with outstanding exploration. Last, the forgetting ability, which causes a particle to overlook some dimensions from its exemplars, can reduce correlation between variables to some extent.

5.2.2. Multimodal functions (f_6-f_{20})

In the second set of experiments, 15 multimodal functions are tested. The comparison results presented in Table 3 indicate that F_PSO and DEPSO do

Table 2: Comparison results on 5 unimodal functions of CEC'13 test suite

Algorithms	f_1				f_2				f_3				f_4				f_5			
	Mean	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)
XPSO	3.79E-13	1.62E-13	2.27E-13	12.96	4.22E+06	2.63E+06	3.22E+06	14.80	1.80E+06	1.94E+06	5.14E+05	14.11	1.79E+02	1.09E+02	1.35E+02	13.78	4.21E-13	1.21E-13	3.41E-13	13.76
F_PSO	1.44E-13	1.06E-13	2.27E-13	12.63	5.20E+06	1.14E+06	5.08E+06	14.11	6.35E+07	8.79E+07	1.40E+07	14.54	5.84E+03	1.25E+03	5.38E+03	13.65	1.14E-13	0.00E+00	1.14E-13	12.88
DEPSO	4.18E+02	6.72E+02	2.27E-13	11.09	2.23E+06	2.40E+06	1.62E+06	16.74	2.39E+09	4.43E+09	5.29E+08	16.07	9.14E+02	8.91E+02	6.32E+02	13.51	1.28E+02	3.99E+02	4.62E+01	11.36
OLPSO	6.06E-14	1.02E-13	0.00E+00	8.91	4.40E+07	1.86E+07	3.91E+07	9.75	3.04E+10	1.33E+10	2.88E+10	9.70	1.35E+05	2.25E+04	1.41E+05	9.00	1.29E-13	3.93E-14	1.14E-13	8.82
PSODDS	8.45E+01	1.46E+02	6.82E-13	20.05	1.16E+06	1.30E+06	3.29E+05	22.62	7.78E+09	7.80E+09	3.05E+09	23.04	5.66E+03	4.03E+03	3.55E+03	22.29	6.11E+01	6.72E+01	5.81E-08	20.20
CCPSO-ISM	3.78E-11	6.58E-11	4.55E-13	12.53	5.74E+06	3.30E+06	5.87E+06	13.72	8.87E+06	9.91E+06	2.49E+06	13.57	2.23E+02	1.26E+02	1.66E+02	13.38	3.74E-12	3.53E-12	1.59E-12	11.86
SRPSO	1.31E+02	2.01E+02	6.82E-13	14.95	2.94E+06	2.22E+06	1.62E+06	16.86	3.74E+09	4.67E+09	3.22E+08	17.22	2.11E+02	7.89E+01	1.74E+02	15.95	1.14E+02	1.48E+02	9.09E-13	16.02
HCLPSO	4.60E-13	1.43E-13	2.27E-13	12.28	3.34E+06	5.94E+05	1.77E+06	12.29	3.63E+07	3.50E+07	1.90E+07	13.30	2.45E+03	7.69E+02	2.22E+03	12.25	4.36E-13	7.74E-14	3.41E-13	12.02
GLPSO	4.55E-13	8.91E-13	0.00E+00	15.84	3.37E+06	1.39E+06	3.17E+06	16.21	1.60E+07	1.26E+07	1.10E+07	16.35	1.38E+03	8.54E+02	9.37E+02	15.73	1.07E-13	1.28E-14 15.25	1.14E-13	15.64
EPSO	3.91E-13	6.09E-14	2.27E-13	14.53	1.73E+06	5.20E+05	1.61E+06	20.03	1.05E+08	1.33E+08	2.43E+07	18.20	2.06E+02	1.17E+02	1.52E+02	18.07	4.77E-13	3.91E-13	4.21E-13	14.20

much worse than XPSO and GLPSO, measured by the number of attained best results on mean values, mainly due to that the later ones take advantage of much helpful historical knowledge from different exemplars, which is beneficial for maintaining the population diversity. For *Rastrigin* (f_{11}), HCLPSO and EPSO significantly dominate other algorithms. Moreover, GLPSO and OLPSO can find out the global optimum value on this problem in a few runs. The results further verify that extracting useful information from different exemplars is very efficient in solving problems with huge number of local optima. However, when f_{11} is extended to a non-separable problem, i.e., *rotated Rastrigin* (f_{12}), the performance of HCLPSO dramatically deteriorate. On the contrary, XPSO gives very stable performance on the two functions. More important, XPSO dominates the other 9 algorithms on f_{12}, in terms of both mean value and median value. Note that all the selected algorithms have similar results on many multimodal functions, i.e., f_6, f_8, f_{15}, f_{18}, and f_{20}, in terms of the order of the mean values. Generally, among those outstanding algorithms, XPSO has the most competitive performance since it achieves the best mean values on 6 out of the 15 multimodal problems, followed by GLPSO, HCLPSO, and EPSO.

Table 3: Comparison results on 15 multimodal functions of CEC'13 test suite

Algorithms	f_6				f_7				f_8				f_9			
	Mean	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)
XPSO	5.83E+01	2.25E+01	5.26E+01	14.40	3.83E+00	2.70E+00	2.50E+00	16.34	2.09E+01	2.65E-02	2.09E+01	16.11	9.81E+00	2.13E+00	9.04E+00	35.02
F_PSO	3.51E+01	2.51E+01	1.53E+01	13.65	3.53E+01	1.21E+01	3.63E+01	17.20	2.09E+01	4.97E-02	2.09E+01	16.49	1.54E+01	2.36E+00	1.60E+01	37.96
DEPSO	7.58E+01	5.60E+01	6.49E+01	12.01	5.09E+01	3.97E+01	4.30E+01	16.58	2.10E+01	4.88E-02	2.10E+01	15.07	1.54E+01	3.68E+00	1.61E+01	47.39
OLPSO	4.89E+01	6.78E+00	4.87E+01	9.30	1.71E+02	4.39E+01	1.67E+02	12.97	2.10E+01	8.53E-02	2.10E+01	12.46	2.40E+01	2.46E+00	2.35E+01	38.13
PSODDS	6.05E+01	2.95E+01	6.48E+01	20.14	1.05E+02	1.98E+01	1.05E+02	25.48	2.09E+01	4.61E-02	2.10E+01	25.44	2.61E+01	2.26E+00	2.62E+01	43.60
CCPSO-ISM	6.37E+01	2.32E+01	6.07E+01	12.74	6.80E+00	3.53E+00	5.25E+00	18.80	2.09E+01	4.06E-02	2.09E+01	17.60	1.24E+01	2.21E+00	1.18E+01	42.89
SRPSO	6.95E+01	1.90E+01	6.92E+01	16.40	4.44E+01	1.96E+01	3.77E+01	19.25	2.10E+01	2.91E-02	2.10E+01	18.35	1.42E+01	2.68E+00	1.35E+01	39.55
HCLPSO	3.75E+01	1.58E+01	2.64E+01	12.46	2.51E+01	4.89E+00	1.90E+01	16.11	2.09E+01	3.60E-02	2.10E+01	16.32	2.17E+01	3.35E+00	2.31E+01	38.89
GLPSO	2.12E+01	1.27E+01	1.36E+01	15.64	2.51E+01	1.26E+01	2.13E+01	18.71	2.09E+01	6.47E-02 18.05	2.09E+01	33.75	1.59E+01	2.57E+00	1.63E+01	16.34
EPSO	2.17E+01	5.83E+00	1.30E+01	19.75	3.23E+01	8.99E+00	2.93E+01	21.78	2.09E+01	4.69E-02	2.10E+01	23.15	2.24E+01	2.76E+00	2.28E+01	42.66
Algorithms	f_{10}				f_{11}				f_{12}				f_{13}			
	Mean	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)
XPSO	1.02E-01	4.89E-02	8.50E-02	13.56	1.82E+01	3.81E+00	1.69E+01	14.87	3.89E+01	1.98E+01	2.89E+01	15.48	1.06E+02	5.10E+01	9.20E+01	16.51
F_PSO	1.99E-01	6.18E-02	1.85E-01	13.93	5.19E+01	1.24E+01	5.31E+01	14.62	1.72E+02	7.59E+00	1.72E+02	15.59	1.75E+02	8.55E+00	1.77E+02	15.39
DEPSO	1.24E+02	1.01E+02	1.17E+02	15.89	3.44E+01	1.08E+01	3.34E+01	13.07	1.24E+02	6.37E+01	1.57E+02	14.20	1.69E+02	2.79E+01	1.73E+02	13.66
OLPSO	1.61E+02	7.34E+01	1.68E+02	10.63	2.47E+00	4.28E+00	0.00E+00	11.07	1.45E+02	5.95E+01	1.22E+02	11.94	2.11E+02	3.16E+01	2.10E+02	11.89
PSODDS	7.63E+01	4.89E+01	7.62E+01	21.10	7.98E+01	2.17E+01	7.39E+01	19.99	1.62E+02	3.56E+01	1.60E+02	20.31	2.71E+02	4.49E+01	2.66E+02	21.52
CCPSO-ISM	1.18E-01	4.60E-02	1.05E-01	14.66	2.39E+01	6.32E+00	2.29E+01	14.54	3.91E+01	1.34E+01	3.23E+01	16.42	1.02E+02	4.42E+01	9.21E+01	17.06
SRPSO	9.07E+01	6.78E+01	6.53E+01	17.45	3.86E+01	8.66E+00	3.78E+01	17.61	5.60E+01	1.46E+01	5.19E+01	18.16	1.19E+02	2.15E+01	1.22E+02	17.26
HCLPSO	1.90E-01	1.13E-01	2.91E-01	13.08	1.03E-13	2.87E-14	1.14E-13	13.95	5.49E+01	1.34E+01	5.17E+01	14.39	1.46E+02	1.07E+01	1.45E+02	13.88
GLPSO	1.32E-01	5.47E-02	1.18E-01	16.72	7.96E-02	1.46E-01	0.00E+00	17.47	4.77E+01	1.51E+01	4.68E+01	17.45	1.12E+02	2.64E+01	1.03E+02	16.84
EPSO	1.53E-01	7.15E-02	1.27E-01	16.40	1.16E-13	2.59E-14	1.14E-13	16.65	5.91E+01	1.41E+01	5.87E+01	19.30	1.17E+02	3.01E+01	1.12E+02	19.27
	f_{14}				f_{15}				f_{16}				f_{17}			
	Mean	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)
XPSO	1.39E+03	4.35E+02	1.47E+03	15.35	5.27E+03	1.72E+03	6.27E+03	16.67	2.40E+00	2.44E-01	2.45E+00	19.57	8.16E+01	4.43E+01	5.27E+01	14.93
F_PSO	2.73E+03	5.35E+02	2.83E+03	14.32	6.19E+03	3.08E+02	6.30E+03	14.71	2.47E+00	2.01E-01	2.50E+00	17.99	1.70E+02	9.87E+00	1.71E+02	13.77
DEPSO	1.56E+03	5.32E+02	1.56E+03	13.24	4.36E+03	1.64E+03	3.82E+03	13.58	2.43E+00	4.03E-01	2.52E+00	19.05	6.72E+01	3.18E+01	5.99E+01	12.41
OLPSO	8.56E+01	9.27E+01	6.27E+01	11.14	7.32E+03	1.17E+03	7.82E+03	11.14	3.50E+00	4.04E-01	3.46E+00	15.39	3.10E+01	2.99E-01	3.09E+01	10.25
PSODDS	2.00E+03	4.55E+02	1.84E+03	20.51	4.09E+03	3.99E+02	4.10E+03	21.64	8.72E-01	4.15E-01	6.32E-01	23.67	1.08E+02	1.48E+01	1.09E+02	20.91
CCPSO-ISM	1.40E+03	3.17E+02	1.24E+03	15.91	3.67E+03	1.28E+03	3.25E+03	16.37	2.39E+00	2.33E-01	2.44E+00	20.53	5.85E+01	6.96E+00	5.82E+01	14.41
SRPSO	1.37E+03	2.20E+02	1.34E+03	16.75	3.02E+03	6.53E+02	2.88E+03	16.58	2.50E+00	2.32E-01	2.54E+00	19.57	6.16E+01	7.09E+00	6.24E+01	16.16
HCLPSO	1.74E+01	2.81E+01	3.32E-01	14.06	3.99E+03	2.98E+02	4.09E+03	13.36	2.23E+00	1.20E-01	1.58E+00	19.04	3.87E+01	8.38E-02	3.96E+01	13.00
GLPSO	3.32E+00	1.89E+00	2.54E+00	19.55	3.81E+03	4.59E+02	3.83E+03	16.30	4.89E-01	2.28E-01	4.36E-01	19.55	3.20E+01	4.38E-01	3.20E+01	16.30
EPSO	3.17E+01	1.04E+01	3.07E+01	21.41	3.61E+03	4.01E+02	3.76E+03	23.35	2.68E+00	2.88E-01	2.67E+00	28.50	3.65E+01	2.32E+00	3.63E+01	15.82
	f_{18}				f_{19}				f_{20}							
	Mean	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)				
XPSO	1.93E+02	8.48E+00	1.94E+02	17.58	2.58E+00	5.18E-01	2.48E+00	14.46	1.18E+01	1.20E+00	1.12E+01	16.26				
F_PSO	1.98E+02	7.37E+00	2.00E+02	14.54	1.21E+01	7.81E-01	1.21E+01	13.35	1.27E+01	7.30E-01	1.23E+01	14.81				
DEPSO	2.18E+02	1.50E+01	2.19E+02	12.24	5.63E+00	1.21E+01	2.78E+00	12.23	1.25E+01	1.22E+00	1.21E+01	11.73				
OLPSO	2.36E+02	2.00E+01	2.42E+02	10.93	2.69E+00	5.20E-01	2.65E+00	9.75	1.44E+01	6.02E-01	1.45E+01	10.54				
PSODDS	1.71E+02	3.18E+01	1.65E+02	20.48	1.02E+01	5.16E+00	8.17E+00	19.39	1.42E+01	8.48E-01	1.45E+01	20.09				
CCPSO-ISM	1.90E+02	1.65E+01	1.92E+02	14.98	2.67E+00	4.17E-01	2.53E+00	13.65	1.23E+01	1.64E+00	1.18E+01	15.70				
SRPSO	1.99E+02	2.02E+01	2.08E+02	16.53	4.05E+00	1.81E+00	3.03E+00	15.20	1.39E+01	1.29E+00	1.50E+01	16.82				
HCLPSO	1.94E+02	2.45E+01	1.21E+02	14.05	1.88E+00	2.20E-01	1.33E+00	12.87	1.22E+01	6.29E-01	1.19E+01	13.11				
GLPSO	1.94E+02	1.14E+01	1.79E+02	16.81	2.70E+00	2.17E-01	2.70E+00	15.79	1.24E+01	1.55E+00	1.15E+01	16.39				
EPSO	1.94E+02	1.39E+01	1.59E+02	16.06	2.84E+00	2.73E-01	2.86E+00	16.33	1.20E+01	1.47E+01	1.12E+01	22.56				

5.2.3. Composition functions (f_{21}-f_{28})

In the third set of experiments, 8 composition functions are explored. From the comparison results listed in Table 4 we can see that XPSO yields the best performance on 3 out of the 8 composition problems, i.e., f_{22}, f_{27}, and f_{28}, contributing to high accuracy of mean value and median value. On the other hand, GLPSO gives the best results on f_{24} while EPSO yields the most outstanding performance on f_{26}. Note that 7 out of the 10 peer algorithms offer the same median value on f_{28} though XPSO yields the fared marginally better mean value on the function. Based on these observations, it may be inferred that XPSO has outstanding performance on complicated composition functions. Note that there is no algorithm can find out the global optimum for a composition function. The unsatisfactory outcome manifests that how to insight characteristics within complicated problems, and then to take full advantages of it are still crucial problems in EAs domain.

Table 4: Comparison results on 8 composition functions of CEC'13 test suite

Algorithms	f_{21}				f_{22}				f_{23}				f_{24}			
	Mean	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)
XPSO	3.10E+02	8.01E+01	3.00E+02	20.71	1.02E+02	2.82E+01	1.01E+02	20.96	4.79E+03	2.09E+03	6.01E+03	21.57	2.43E+02	1.84E+01	2.52E+02	42.24
F_PSO	3.09E+02	4.58E+01	3.00E+02	19.53	1.66E+03	5.18E+02	1.49E+03	21.60	6.79E+03	3.47E+02	6.71E+03	21.62	2.50E+02	2.07E+01	2.53E+02	43.72
DEPSO	3.28E+02	9.50E+01	3.00E+02	18.64	1.51E+03	4.07E+02	1.51E+03	21.53	4.37E+03	1.34E+03	3.82E+03	23.58	2.56E+02	1.28E+01	2.57E+02	59.53
OLPSO	2.23E+02	6.50E+01	2.00E+02	17.31	3.73E+02	1.99E+02	3.11E+02	17.88	7.55E+03	1.30E+03	8.25E+03	19.35	2.74E+02	5.70E+00	2.74E+02	47.17
PSODDS	3.42E+02	8.51E+01	3.00E+02	24.76	2.35E+03	4.31E+02	2.56E+03	25.76	4.98E+03	7.14E+02	5.00E+03	26.52	2.76E+02	6.36E+00	2.77E+02	45.68
CCPSO-ISM	3.05E+02	5.51E+01	3.00E+02	19.62	1.25E+03	3.43E+02	1.14E+03	21.35	3.34E+03	9.50E+02	2.81E+03	26.87	2.54E+02	1.36E+01	2.57E+02	50.26
SRPSO	3.44E+02	7.94E+01	3.00E+02	22.22	1.64E+03	3.63E+02	1.58E+03	22.90	3.37E+03	5.86E+02	3.35E+03	24.74	2.60E+02	1.12E+01	2.62E+02	45.98
HCLPSO	3.08E+02	4.66E+01	3.00E+02	18.94	1.10E+02	1.05E+01	1.13E+02	19.15	4.57E+03	4.16E+02	4.28E+03	20.34	2.43E+02	6.19E+00	2.55E+02	48.03
GLPSO	3.14E+02	4.66E+01	3.00E+02	20.52	1.07E+02	2.19E+01	1.17E+02	21.35	3.93E+03	5.82E+02	4.07E+03	22.44	2.37E+02	1.01E+01	2.38E+02	38.43
EPSO	2.26E+02	4.23E+01	2.00E+02	22.31	1.48E+02	2.28E+01	1.53E+02	27.69	4.00E+03	5.46E+02	3.88E+03	30.71	2.47E+02	8.14E+00	2.48E+02	44.82

	f_{25}				f_{26}				f_{27}				f_{28}			
	Mean	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)	*Mean*	*Std.Dev*	*Median*	*Time* (s)
XPSO	2.80E+02	7.22E+00	2.81E+02	41.86	2.91E+02	4.85E+01	3.20E+02	42.88	4.81E+02	1.34E+02	4.84E+02	42.27	2.73E+02	4.62E+01	3.00E+02	23.87
F_PSO	2.62E+02	2.22E+01	2.75E+02	46.99	3.12E+02	3.73E+01	3.30E+02	48.43	6.04E+02	8.28E+01	5.92E+02	47.91	2.80E+02	3.60E+01	3.00E+02	25.61
DEPSO	2.76E+02	8.00E+00	2.75E+02	51.98	3.26E+02	5.11E+01	3.44E+02	58.75	7.65E+02	1.01E+02	7.89E+02	57.06	9.77E+02	5.77E+02	1.14E+03	27.86
OLPSO	2.85E+02	5.79E+00	2.85E+02	47.56	2.34E+02	5.96E+01	2.07E+02	52.27	9.54E+02	5.46E+01	9.63E+02	51.36	1.43E+03	2.62E+02	1.43E+03	24.23
PSODDS	2.99E+02	7.89E+00	2.98E+02	47.26	2.43E+02	6.32E+01	2.00E+02	52.78	9.99E+02	9.51E+01	1.00E+03	50.63	1.10E+03	4.79E+02	1.22E+03	30.99
CCPSO-ISM	2.82E+02	7.90E+00	2.83E+02	47.32	2.87E+02	5.80E+01	3.25E+02	54.26	6.82E+02	9.59E+01	7.14E+02	53.13	4.41E+02	2.45E+02	3.00E+02	26.80
SRPSO	2.89E+02	1.06E+01	2.87E+02	45.46	2.79E+02	6.79E+01	3.27E+02	48.57	7.13E+02	9.20E+01	7.27E+02	47.97	6.53E+02	4.14E+02	3.00E+02	26.06
HCLPSO	2.68E+02	1.12E+01	2.74E+02	47.99	2.63E+02	2.89E-02	2.00E+02	52.31	6.15E+02	1.44E+02	5.74E+02	51.30	3.00E+02	9.79E-11	3.00E+02	24.79
GLPSO	2.64E+02	5.85E+00	2.65E+02	39.36	2.63E+02	6.56E+01	2.00E+02	41.38	6.78E+02	8.51E+01	7.03E+02	40.94	3.54E+02	1.26E+02	3.00E+02	25.21
EPSO	2.84E+02	7.04E+00	2.86E+02	49.62	2.08E+02	1.56E+01	2.00E+02	46.58	7.27E+02	1.48E+02	7.93E+02	47.50	3.00E+02	2.49E+01	3.00E+02	26.23

5.3. Comparison on CPU time

The comparison results of CPU times presented in Table 2 - Table 4 indicate that OLPSO displays more promising performance than other peer algorithms while PSODDS and EPSO manifest unfavorable results on these unimodal functions. XPSO, F_PSO, DEPSO, and CCPSO-ISM achieve almost the same time usage results on these functions. The results in Table 3 show that OLPSO also has the best performance while XPSO also displays moderate performance. However, for those complicated composition functions, i.e., f_{24}-f_{28}, all the peer algorithms offer very similar performance. From the results we can see that, in simple functions, more computational time is allocated for the sort and distance computation operators in XPSO while the evaluation time is very short. On the contrary, for complicated problems, the bottleneck of XPSO is the evaluation time for the problems rather than the sort and distance computation operators.

5.4. Significance statistical test of experimental results

To furthermore illustrate the comprehensive performance of all the peer algorithm, and propose an overview of all the peer algorithms, the Friedman test with a significance level of α=0.05 is conducted in this part, the results of which are listed in Table 5. From Table 5, we can see that XPSO attains the best comprehensive performance on CEC'13 test suite, followed by GLPSO, HCLPSO, EPSO, and CCPSO_ISM. In addition, XPSO also achieves the most favorable results on the 5 unimodal and 15 multimodal functions. On the contrary, GLPSO yields the most favorable result on the 8 composition problems, on which XPSO displays the third best performances.

Furthermore, a graphical approach is used to show critical difference (CD) with Friedman rankings, the result of which is demonstrated in Fig. 4. The result indicates that XPSO does not significantly different with GLPSO, HCLPSO, EPSO, and CCPSO_ISM, though it offers the best performance, in terms of the Friedman ranking values.

Table 5: Friedman test of mean values on the CEC'13 test suite

Average Rank	Algorithm	Ranking	Unimodal: f_1-f_5		Multimodal: f_6-f_{20}		Composition: f_{21}-f_{28}	
			Algorithm	Ranking	Algorithm	Ranking	Algorithm	Ranking
1	XPSO	3.59	XPSO	3.20	XPSO	3.50	HCLPSO	3.68
2	GLPSO	3.70	EPSO	4.00	CCPSO_ISM	3.57	GLPSO	3.68
3	HCLPSO	4.05	GLPSO	4.20	GLPSO	3.60	XPSO	3.77
4	EPSO	4.21	HCLPSO	5.40	HCLPSO	3.87	EPSO	4.09
5	CCPSO_ISM	4.41	FPSO	5.20	EPSO	4.40	CCPSO_ISM	4.86
6	F_PSO	6.32	CCPSO_ISM	5.80	SRPSO	6.47	FPSO	4.86
7	SRSPO	6.64	SRPSO	6.60	FPSO	7.27	OLPSO	7.09
8	OLPSO	7.18	OLPSO	6.80	PSODDS	7.30	SRPSO	7.36
9	DEPSO	7.30	PSODDS	6.80	DEPSO	7.50	DEPSO	7.64
10	PSODDS	7.59	DEPSO	7.00	OLPSO	7.53	PSODDS	7.95

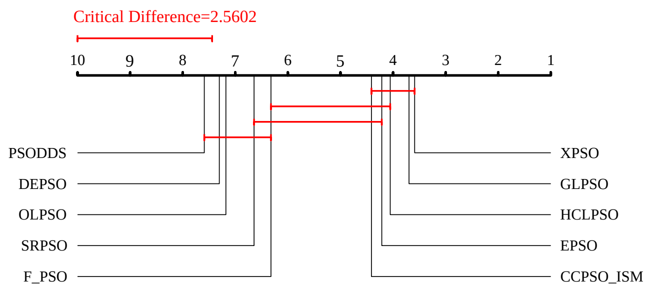


Figure 4: Critical difference with Friedman rankings.

5.5. *Comparisons with other evolutionary algorithms*

In this section, we further compare XPSO with other evolutionary algorithms, i.e., self-adaptive differential evolutionary (SaDE) [33], evolution strategy with covariance matrix adaptation (CMAES) [13], and population-based incremental learning (PBILc) [9], which belong to DE, ES, and estimation distribution algorithm (EDA) communities, respectively. XPSO and the three competitors share a common feature that the statistical results of historical experiences of elites are used to generate promising offspring. Parameter configurations of the new selected peer algorithms are the same as the corresponding references.

The comparison results of independent 30 runs are listed in Table 6. Moreover, the t-test and the Friedman test are also carried out, the results of which are presented at the bottom of Table 6. Note that the symbols "(+)", "(-)", and "(=)" denote that XPSO is significantly better than, significantly worse than, and almost the same as the corresponding competitors respectively, in terms of t-test results. The results listed in the second-to-last row denote the total number of functions on which XPSO is significantly better than, significantly worse than, and almost the same as the corresponding algorithm. For instance, "12/6/10" in the second-to-last row means that XPSO is significantly better than, significantly worse than, and almost the same as SaDE on 12, 6, and 10 test functions, respectively. The rank values of the Friedman test (the smaller the better) are presented at the last row of Table 6.

Table 6: Comparisons between XPSO and other three EAs on CEC'13 test suite

	XPSO	SaDE	CMAES	PBILc
f_1	3.79E-13±1.62E-13	0.00E+00 ±0.00E+00 (-)	4.40E-13±8.49E-14 (=)	7.58E-14±1.01E-13 (=)
f_2	4.22E+06±2.63E+06	4.17E+05±1.72E+05 (-)	4.47E-13 ±8.79E-14 (-)	2.28E+06±1.07E+06 (-)
f_3	1.80E+06±1.94E+06	2.37E+07±2.11E+07 (+)	8.90E+02 ±1.62E+03 (-)	3.25E+06±5.52E+06 (+)
f_4	1.79E+02±1.09E+02	2.97E+03±1.28E+03 (+)	4.40E-13 ±9.90E-14 (-)	3.96E+04±1.14E+04 (+)
f_5	4.21E-13±1.21E-13	0.00E+00 ±0.00E+00 (-)	7.77E-13±5.68E-13 (+)	9.03E-01±1.74E+00 (+)
f_6	5.83E+01±2.25E+01	3.16E+01±2.54E+01 (=)	1.76E+00 ±3.29E+00 (-)	3.21E+01±8.20E+00 (-)
f_7	3.83E+00±2.70E+00	2.58E+01±8.21E+00 (+)	1.41E+01±6.78E+00 (+)	1.82E+00 ±1.45E+00 (=)
f_8	2.09E+01 ±2.65E-02	2.09E+01 ±3.58E-02 (+)	2.15E+01±7.49E-02 (+)	2.09E+01 ±3.17E-02 (=)
f_9	9.81E+00 ±2.13E+00	1.65E+01±1.79E+00 (+)	4.30E+01±5.98E+00 (+)	1.22E+01±7.17E+00 (+)
f_{10}	1.02E-01±4.89E-02	3.08E-01±1.23E-01 (+)	3.36E-02 ±1.86E-02 (-)	5.01E-03±4.68E-03 (-)
f_{11}	1.82E+01±3.81E+00	2.32E-01 ±3.56E-01 (-)	9.44E+01±9.16E+01 (+)	4.21E+01±1.89E+01 (+)
f_{12}	3.89E+01 ±1.98E+01	5.00E+01±1.20E+01 (+)	8.69E+02±9.90E+02 (+)	1.60E+02±6.06E+00 (+)
f_{13}	1.06E+02 ±5.10E+01	1.08E+02±1.32E+01 (+)	1.73E+03±1.04E+03 (+)	1.57E+02±6.94E+00 (+)
f_{14}	1.39E+03±4.35E+02	9.09E-01 ±9.53E-01 (-)	5.59E+03±6.91E+02 (+)	1.43E+03±4.50E+02 (=)
f_{15}	5.27E+03±1.72E+03	4.76E+03±8.24E+02 (-)	5.25E+03±6.23E+02 (-)	1.33E+03 ±3.46E+02 (-)
f_{16}	2.40E+00±2.44E-01	2.28E+00±2.10E-01 (=)	7.30E-02 ±3.08E-02 (-)	2.44E+00±3.39E-01 (=)
f_{17}	8.16E+01±4.43E+01	3.05E+01 ±2.09E-02 (-)	4.07E+03±6.66E+02 (+)	1.89E+02±5.49E+00 (+)
f_{18}	1.93E+02±8.48E+00	1.24E+02 ±3.65E+01 (-)	3.99E+03±6.21E+02 (+)	1.95E+02±6.29E+00 (+)
f_{19}	2.58E+00 ±5.18E-01	4.25E+00±6.63E-01 (+)	3.44E+00±8.03E-01 (+)	3.38E+00±5.55E-01 (=)
f_{20}	1.18E+01±1.20E+00	1.15E+01 ±1.12E+00 (=)	1.28E+01±3.88E-01 (+)	1.31E+01±1.61E+00 (+)
f_{21}	3.10E+02 ±8.01E+01	3.20E+02±7.41E+01 (=)	3.12E+02±7.02E+01 (=)	3.19E+02±4.98E+01 (=)
f_{22}	1.02E+02 ±2.82E+01	1.24E+02±5.32E+01 (=)	7.24E+03±6.77E+02 (+)	1.20E+03±2.89E+02 (+)
f_{23}	4.79E+03±2.09E+03	4.92E+03±8.87E+02 (+)	6.67E+03±7.19E+02 (+)	1.24E+03 ±3.12E+02 (-)
f_{24}	2.43E+02±1.84E+01	2.27E+02±4.08E+00 (-)	7.44E+02±5.56E+02 (+)	2.18E+02 ±7.03E+00 (-)
f_{25}	2.80E+02±7.22E+00	2.66E+02±8.06E+00 (=)	3.40E+02±1.45E+02 (+)	2.44E+02 ±4.67E+00 (-)
f_{26}	2.91E+02±4.85E+01	2.03E+02 ±5.28E+00 (-)	5.22E+02±4.10E+02 (+)	2.98E+02±2.68E+01 (+)
f_{27}	4.81E+02 ±1.34E+02	6.10E+02±5.05E+01 (+)	5.68E+02±9.32E+01 (+)	4.90E+02±5.13E+01 (+)
f_{28}	2.73E+02 ±4.62E+01	3.00E+02±0.00E+00 (+)	1.74E+03±2.07E+03 (+)	3.00E+02±2.11E-09 (+)
t-test	+ / = / -	12 / 6 / 10	19 / 2 / 7	14 / 7 / 7
Rank	2.07	2.30	3.11	2.52

We can see that XPSO yields the most promising performance on the test suite, followed by SaDE, PBILc, and CMAES, in terms of the solution accuracy as well as the t-test results. Although CMAES attains the best mean results on 3 out of the 5 unimodal functions, it sacrifices performance on multimodal functions. On the contrary, SaDE and XPSO not only are very efficient in solving the unimodal problems, but also attain very competitive performance on the multimodal functions. Moreover, the rank values of Friedman test demonstrate that XPSO displays the most favorable performance, followed SaDE, PBILc, and CMAES.

The comparison results also verify that taking advantages of knowledge implied in different individuals is beneficial for improving the population's intelligence. However, the large error values in the complicated composition problems also indicate that how to extract more useful information from the experience is also a challenging issue.

6. Conclusions

In some sense, forgetting ability can be interpreted to have survived through a process of natural selection in animals. Furthermore, a common phenomenon in the human's learning process is dynamic selection of multiple exemplars. Therefore, inspired by the animal behaviors, we proposed XPSO in this paper. The implementation of XPSO was given in details, and the sensitivity analysis of new introduced parameters was conducted by extensive experiments. Furthermore, the advantages of XPSO were verified by comparing it with other 9 PSO variants and 3 non-PSO algorithms on the CEC'13 test suite. The comparison results indicate there are few merits of XPSO. Firstly, different forgetting abilities assigned to different particles cause knowledge extracted from a same exemplar vary among the different particles and for a same particle at different evolutionary stages, which is beneficial for exploration. Moreover, the multi-exemplar strategy causes a particle to extract much more useful information in each generation. Lastly, much knowledge of elites, rather than a user's prior knowledge of the parameters settings, is applied to update the three acceleration coefficients aiming to satisfy different requirements of different evolutionary stages.

Although XPSO obtains a competitive performance, there are two problems need to be further studied in our future works. The one is the efficiency and effectiveness of the new introduced parameters in XPSO need to in-depth analysis based on different benchmark functions though preliminary discussion has been proposed in this paper. The other one is how to extract much more helpful knowledge from a population's experience, and then applying it to adaptive adjust acceleration coefficients, and direct individuals to perform a highly efficient search process.

Acknowledgment. This study was funded by the National Natural Science Foundation of China (Grant Nos. 61663009, 61602174, 61762036, 61806204, 61876136), the National Natural Science Foundation of Jiangxi Province (Grant No. 20171BAB202012, 20171BAB202019), the Foundation of Fujian Province Great Teaching Reform (Grant No. FBJG 20180015), and the Research Project of Jiangxi Provincial Department of Communication and Transportation (No. 2017D0038).

Reference

[1] K.G. Akers, A. Martinez-Canabal, L. Restivo *et al.*, Hippocampal neurogenesis regulates forgetting during adulthood and infancy, Science, vol. 344, no. 6184, pp. 598-602, May 2014.

[2] M. Asada, Modeling early vocal development through infant-caregiver interaction: A review, IEEE Trans. Cogn. Dev. Syst. vol. 8, no. 2, pp. 128-138, June 2016.

[3] E.K. Aydoğan, Y. Delice, U. Özcan, C. Gencer, and Ozkan Bali, Balancing stochastic U-lines using particle swarm optimization, J. Intell. Manuf. vol. 30, no. 1, pp. 97-111, Jan. 2019.

[4] A. Baeck, K. Maes, C. Van Meel *et al.*, The transfer of object learning after training with multiple exemplars, Front. Psychol. vol. 7, 1386, Sep. 2016.

[5] J. Barrett, K.J.S. Zollman, The role of forgetting in the evolution and learning of language, J. Exp. Theor. Artif. Intell. vol. 21, no. 4, pp. 293-309, 2009.

[6] E.R. Behrend, A.S. Powers, M.E. Bitterman, Interference and forgetting in bird and fish, Science, vol. 167, no. 3917, pp. 389-390, Jan. 1970.

[7] M. Campos, R.A. Krohling, I. Enriquez, Bare bones particle swarm optimization with scale matrix adaptation, IEEE Trans. Cybern. vol. 44, no. 9, pp. 1567-1578, Sep. 2014.

[8] R. Cheng, Y.C. Jin, A competitive swarm optimizer for large scale optimization, IEEE Trans. Cybern. vol. 45, no. 2, pp. 191-204, Feb. 2015.

[9] A. Ducoulombier, Extending Population-Based Incremental Learning to Continuous Search Spaces, in: Proc. of International Conference on Parallel Problem Solving From Nature, Amsterdam, Netherlands, Sep. 1998, pp. 418-427.

[10] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proc. of 6th International Symposium on Micromachine and Human Science, Nagoya, Japan, 1995, pp.39-43.

[11] Y.J. Gong, J.J. Li, Y.C. Zhou *et al.*, Genetic learning particle swarm optimization, IEEE Trans. Cybern. vol. 46, no. 10, pp. 2277-2290, Oct. 2016.

[12] H.G. Han, W. Lu, J.F. Qiao, An adaptive multiobjective particle swarm optimization based on multiple adaptive methods, IEEE Trans. Cybern. vol. 47, no. 9, pp. 2754-2767, Sep. 2017.

[13] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evol. Comput. vol. 9, no. 2, pp. 159-195, 2001.

[14] B.M. Ivatloo, Combined heat and power economic dispatch problem solution using particle swarm optimization with time varying acceleration coefficients, Electr. Power Syst. Res. vol. 95, no. 1, pp. 9-18, 2015.

[15] X. Jin, Y.Q. Liang, D.P. Tian *et al.*, Particle swarm optimization using dimension selection methods, Appl. Math. Comput. vol. 219, no. 10, pp. 5185-5197, 2013.

[16] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proc. of IEEE International Conference on Neural Network, Perth, Australia, 1995, vol.4, pp.1942-1948.

[17] J. Kennedy, R. Mendes, Neighborhood topologies in fully-informed and best-of-neighborhood particle swarms, in: Proc. of IEEE International Workshop on Soft Computing in Industrial Applications, New York, Jun. 2003, pp. 45-50.

[18] S.H. Leung, C.F. So, Nonlinear RLS algorithm using variable forgetting factor in mixture noise, in: Proc. IEEE Int. Conf. on Acou. 2007, vol. 6, pp. 3777-3780.

[19] C. Li, S. Yang, T.T. Nguyen, A self-learning particle swarm optimizer for global optimization problems, IEEE Trans. Syst. Man, Cybern. B, Cybern. vol. 42, no. 3, pp. 627-646, Jun. 2012.

[20] Y. Li, Z.H. Zhan, S. Lin *et al.*, Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems, Inf. Sci. vol. 293, no. 3, pp. 370-382, 2015.

[21] J. Li, J.Q. Zhang, C.J. Jiang *et al.*, Composite particle swarm optimizer with historical memory for function optimization, IEEE Trans. Cybern. vol. 45, no. 10, pp. 2350-2363, Oct. 2015.

[22] J.J. Liang, P.N. Suganthan, Dynamic multi-swarm particle swarm optimizer, in: Proc. of IEEE Swarm Intelligence Symposium, Pasadena, CA, USA, 2005, pp.124-129.

[23] J.J. Liang, A.K. Qin, P.N. Suganthan *et al.*, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. vol. 10, no. 3, pp. 281-295, Jun. 2006.

[24] J.J. Liang, B.Y. Qu, P.N. Suganthan, Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization, Nanyang Technological Univ., Singapore, Tech. Rep., 2013.

[25] Q. Liu, W. Wei, H. Yuan *et al.*, Topology selection for particle swarm optimization, Inf. Sci. vol. 363, pp. 154-173, 2016.

[26] J. Lu, H. Hu, Y. Bai, Generalized radial basis function neural network based on an improved dynamic particle swarm optimization and adaboost algorithm, Neurocomputing, vol. 152, pp. 305-315, 2015.

[27] N. Lynn, P.N. Suganthan, Ensemble particle swarm optimizer, Appl. Soft Comput. vol. 55, pp. 533-548, 2017.

[28] N. Lynn, M.Z. Ali, P.N. Suganthan, Population topologies for particle swarm optimization and differential evolution, Swarm Evol. Comput. vol. 39, pp. 24-35, 2018.

[29] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning partile swarm optimization with enhanced exploration and exploitation, Swarm Evol. Comput. vol. 24, pp. 11-24, 2015.

[30] A. Marco, d.O. Montes, S. Thomas *et al.*, Frankenstein's PSO: a composite particle swarm optimization algorithm, IEEE Trans. Evol. Comput. vol. 13, no. 5, pp. 1120-1132, Sep. 2009.

[31] D.A. Miller, J.M. Zurada, A dynamical system perspective of structural learning with forgetting, IEEE Trans. Neural Networ. vol. 9, no. 3, pp. 508-515, May 1998.

[32] T. Peram, K. Veeramachaneni, C.K. Mohan, Fitness-distance-ratio based particle swarm optimization, in: Proc. of Swarm Intelligence Symposium, Apr. 2003, pp. 174-181.

[33] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization, IEEE Trans. Evol. Comput. vol. 13, no. 2, pp. 398-417, Apr. 2009.

[34] B.Y. Qu, J.J. Liang, P.N. Suganthan, Niching particle swarm optimization with local search for multi-modal optimization, Inf. Sci. vol. 197, no. 197, pp. 131-143, 2012.

[35] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput. vol. 8, no. 3, pp. 240-255, Jun. 2004.

[36] C. Sandovall, M. Chakraborty, C. MacMullen *et al.*, Scribble scaffolds a signalosome for active forgetting, Neuron. vol. 90, no. 6, pp. 1230-1242, May 2016.

[37] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: Proc. IEEE World Congress on Computational Intelligence, Anchorage, AK, May, 1998, pp. 69-73.

[38] M.R. Tanweer, S. Suresh, N. Sundararajan, Self regulating particle swarm optimization algorithm, Inf. Sci. vol. 294, pp. 182-202, 2015.

[39] K.E. Twomey, S.L. Ranson, J.S. Horst, That's more like it: multiple exemplars facilitate word learning, Infant & Child Dev. vol. 23, no. 2, pp. 105-122, 2014.

[40] H.B. Wang, J. Dong, Y.L. Wang, High-ordere feedback iterative learning control algorithm with forgetting factor, Math. Prob. in Eng. vol. 2015, Article ID 826409, 7 pages

[41] D.S. Wang, D.P. Tan, L. Liu, Particle swarm optimization algorithm: an overview, Soft Compt. vol. 22, no. 2, pp. 387-408, 2018.

[42] G.H. Wu, R. Mallipeddi, P.N. Suganthan, Ensemble strategies for population-based optimization algorithms - A survey, Swarm Evol. Comput. vol. 44, pp. 695-711, 2019.

[43] X.W. Xia, J.N. Liu, Z.B. Hu, An improved particle swarm optimizer based on tabu detecting and local learning strategy in a shrunk search space, Appl. Soft Comput. vol. 23, no. 5, pp. 76-90, 2014.

[44] X.W. Xia, C.W. Xie, B. Wei *et al.*, Particle swarm optimization using multi-level adaptation and purposeful detection operators, Inf. Sci. vol. 385, pp. 174-195, 2017.

[45] B. Xin, J. Chen, J. Zhang *et al.*, Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: A review and taxonomy, IEEE Trans. Syst. Man, Cybern. C, Appl. Rev. vol. 42, no. 5, pp. 744-767, Sep. 2012.

[46] C. Yang, W. Gao, N. Liu *et al.*, Low-discrepancy sequence initialized particle swarm optimization algorithm with nonlinear time-varying inertia weight, Appl. Soft Comput. vol. 29, pp. 386-394, 2015.

[47] K. Yasuda, N. Iwasaki, Adaptive particle swarm optimization using velociy information of swarm, in: Proc. of IEEE International Conference on Systems, Man and Cybernetics, Hague, Netherlands, 2004, pp. 3475-3481.

[48] Z.H. Zhan, J. Zhang, Y. Li *et al.*, Adaptive particle swarm optimization, IEEE Trans. Syst., Man, Cybern. B, Cybern. vol. 39, no. 6, pp. 1362-1381, Dec. 2009.

[49] Z.H. Zhan, Y. Li, Y.H. Shi, Orthogonal learning particle swarm optimization, IEEE Trans. Evol. Comput. vol. 15, no. 6, pp. 832-847, Dec. 2011.

[50] https://medicalxpress.com/news/2011-10-forgetting-is-part-of-remembering.html