Particle swarm optimization using multi-level adaptation and purposeful detection operators

Xuewen Xia*[a,b], Chengwang Xie[a,b], Bo Wei[a,b], Zhongbo Hu[c], Bojian Wang[a,b], Chang Jin[a,b]

[a] *School of Software, East China Jiaotong University, Jiangxi, 330013 China*
[b] *Intelligent Optimization and Information Processing Lab., East China Jiaotong University, Jiangxi, 330013 China*
[c] *School of Information and Mathematics, Yangtze University, Hubei, 434023 China*

Abstract

Particle swarm optimization (PSO) algorithm has shown favorable performance on global optimization problems. However, it is prone to premature convergence since a monotonic and static learning model applied for all particles makes PSO unable to deal with different complex situations. Moreover, there is no efficient operator to help population detect some promising positions purposefully if the population has been trapped into potential local optima. To solve the shortcomings, a sophisticated PSO (SopPSO) algorithm based on multi-level adaptation and purposeful detection is proposed in this research. Relying on the multi-level adaptation, a particle not only updates its neighbors based on its current fitness landscape but also periodically re-selects target dimensions which the particle learns from its neighbors. The multi-level adaptive strategy applied in individual-level and dimension-level enables PSO to have a more accurate simulation on emergent collective behaviors. Furthermore, a purposeful detection operator based on some historical information is proposed to help the population to escape from local optima. In addition, a simple local-searching strategy is introduced to improve the accuracy of elitist particles. A set of experiments has verified the efficiency of each proposed component. At last, extensive experimental studies on CEC'13 test suites illustrate the effectiveness and efficiency of SopPSO.

Keywords: Particle swarm optimization, Multi-level adaptation, Purposeful

detection, Local-learning strategy, Numerical optimization

1. Introduction

In recent years, many real-world problems have become extremely complex and are difficult to solve using conventional optimization algorithms. Since meta-heuristic optimization algorithms have shown more favorable performance on non-convex and non-differentiable problems, they capture many researchers' attention. During the last few years, various meta-heuristic optimization algorithms have been developed and successfully applied to diverse difficult real-life problems [34, 35, 36]. According to motivations, these majority of meta-heuristic optimization algorithms can be broadly categorized into: 1) animal activity-based algorithms [9, 11, 22, 44], 2) human activity-based algorithms [2, 27], and 3) physics-based algorithms [4, 23, 29].

Particle swarm optimization (PSO) [11] inspired by the social cooperative behavior of bird flocking and fish schooling is a typical animal activity-based meta-heuristic optimization algorithm. In recent years, PSO has captured great attention and has been successfully applied in many applications [6, 19, 33, 37]. However, many researches indicate that the canonical PSO suffers from premature convergence when solving complex problems [45]. Therefore, avoiding premature convergence has become the most important and appealing goal in PSO research on multimodal problems.

The most popular approach is to foster the diversity of the population based on local version model PSO or dynamic adjustment for topology. Although these strategies have shown more promising performance on keeping diversity of population, the modifications sacrifice the performance on convergence speed. To balance the contradiction between population diversity and convergency speed, some time-varying parameters [26, 31] are introduced. However, the simple iteration-based tuning strategies may run into the risk of inappropriate adjusting parameters due to the information of evolutionary state inappropriately utilized. In recent years, to bring more intelligence to PSO, some

adaptive techniques are introduced to tune parameters [43, 45, 48] and learning patterns [14, 20]. Although some aforementioned strategies overcome premature convergence to some degree, it is unavoidable that the population may fall into a local optimum when optimizing a difficult multimodal function. Thus, some genetic operators, e.g., crossover operator and mutation operator [8, 39], are proposed to help population jump out of the local optimum. Nevertheless, these operators give the population few purposeful searching directions.

Based on these previous researches, a sophisticated PSO (SopPSO) is proposed in this study, in which some new features are introduced to improve its comprehensive performance. Firstly, a multi-level adaptation of learning model based on particles' fitness landscape is applied both on individual level and dimensional level. Relying on the strategy, a particle can not only adjust its neighbors but also re-select target dimensions on which the particle learns from its neighbors. Secondly, a purposeful detecting operator is introduced to guide elite individuals to detect a more promising position, and then prompts the population to jump out of local optimum traps. Thirdly, a simple local-searching strategy is proposed to enhance the exploitation capability of SopPSO.

The rest of this paper is organized as follows. Section 2 describes the framework of the canonical PSO and reviews some PSO modifications. The details of SopPSO are described in Section 3. The performance of parameters and components in SopPSO are detailed in Section 4, and the experimental results between SopPSO and other 13 state-of-art EAs and discussions are introduced in Section 5. Finally, conclusions are given in Section 6.

2. Particle swarm optimization

In PSO, the i^{th} particle is represented by a position vector $\mathbf{X}_i = [x_{i1}, x_{i2}, ..., x_{iD}]$ and a velocity vector $\mathbf{V}_i = [v_{i1}, v_{i2}, ..., v_{iD}]$, where D represents the dimensionality of a solution space. The vector $\mathbf{X}_i$ is regarded as a candidate solution to a problem while the vector $\mathbf{V}_i$ is treated as a searching direction and step size of the i^{th} particle. Along with the optimization process, each particle adjusts its trajectory relying on two vectors, named as its personal historical

best position $\mathbf{Pb}_i = [pb_{i1}, pb_{i1}, ..., pb_{iD}]$ and its neighbors' best-so-far position $\mathbf{Nb}_i = [nb_{i1}, nb_{i1}, ..., nb_{iD}]$, respectively. The updating rules of a particle's velocity and position are defined as (1) and (2), respectively.

$$v_{ij}^{t+1} = w \cdot v_{ij}^{t} + c_1 \cdot r_1 \cdot (pb_{ij} - x_{ij}^{t}) + c_2 \cdot r_2 \cdot (nb_{ij} - x_{ij}^{t}) \tag{1}$$

$$x_{ij}^{t+1} = x_{ij}^{t} + v_{ij}^{t+1} \tag{2}$$

where w represents an inertia weight that determines how much the previous velocity is preserved; c_1 and c_2 are known as two acceleration coefficients that prescribe the relative learning weights for $\mathbf{Pb}_i$ and $\mathbf{Nb}_i$, which are called "self-cognitive" and "social-learning", respectively; r_1 and r_2 are two random numbers generated in the interval [0, 1]; x_{ij}^{t} and v_{ij}^{t} represent the i^{th} particle's position and velocity in the j^{th} dimension in generation t, respectively.

2.1. PSO variants

2.1.1. Parameter adjustment

Many works indicate that a larger w facilitates the global searching capability while a smaller one is beneficial for the local-searching capability. So it seems natural to select a time-varying w to offset the contradiction between the exploration ability and the exploitation ability of PSO. The most ubiquitous update rule of w, introduced by Shi and Eberhart [31] in 1998, is linearly decreasing from 0.9 to 0.4 during the evolutionary process. However, the linearly decreasing strategy does not truly reflect the actual searching process since the searching process of PSO is nonlinear and complicated.

Thus, some nonlinearly-varying strategies [5, 32] are proposed to tune w. The simulation results illustrate that these nonlinearly-varying adjustment can improve the performance of PSO to some extent. Unlike monotonically decreasing strategies in [5, 31, 32], an oscillating adjustment strategy proposed by Kentzoglanakis [13] adopts a sinusoidal function to regulate w oscillating between a predefined upper bound and lower bound. The periodically oscillating w promotes and suspends particle momentum, the purposes of which are to

drive the swarm towards global and local search, respectively. Motivated by the time-varying w, Ratnaweera *et al.* further proposed a self-organizing H-PSO with time-varying acceleration coefficients, i.e., c_1 and c_2, in [26].

Although these iteration-based strategies can improve PSO's performance in various degrees, they may run into risk of inappropriate tuning parameters since the information of evolutionary state is not appropriately utilized. To layout a more satisfactory adjusting method, Zhan [45] utilizes population's distribution and particles' fitness to carry out an evolutionary state estimation (ESE), based on which an adaptive method for tuning w, c_1, and c_2 is proposed.

2.1.2. Population topology adjustment

Although much research suggests that the population topology in PSO with fewer connections and more connections might perform better on multimodal problems and unimodal problems [10, 12, 32] respectively, it is very difficult for us to fix a proper neighbor structure for a specific problem in advance since many real applications are black-box problems. Therefore, quite a few dynamical adjustments for topology have been proposed in recent years [16, 17, 24, 47].

In [24], Veeramachaneni proposed a fitness-distance-ration-based PSO (FDR) in which Euclidian distance and fitness are deemed as criteria for selecting exemplars for a specific particle. Liang [17] proposed a comprehensive learning particle swarm optimization (CLPSO), in which a particle chooses different particles' historical information to update its velocity. The new strategy makes the particles have more exemplars to learn from and a larger potential space to fly through, which give CLPSO a promising performance on multimodal problems. In [16, 47], a dynamic multi-swarm PSO (DMSPSO) was proposed in which a larger population is divided into many small-sized subswarms that can be regrouped after a predefined regrouping period. The dynamic adjustment technique not only keeps population's diversity but also enables good information obtained by a subswarm to be periodically exchanged with other subswarms.

In addition, to endow the population with more intelligence for dealing with different complex situations, a self-learning particle swarm optimizer (SLPSO)

was proposed [14]. In SLPSO, each particle has four candidate learning models (i.e., population topologies). At each generation, each individual can adopt a proper learning model relying on its local fitness landscape to carry out the search process. The experiments manifest that the four different learning models enable each particle to independently deal with various situations, and the adaptive learning framework at individual level can improve the local-searching ability and overcome the premature phenomena of PSO to some extent.

2.1.3. Hybridization strategy

Since different evolutionary algorithms (EAs) and operators have their own merits, it seems natural to think of integrating different algorithms or operators to handle complex problems. For example, three genetic operators, i.e., selection [1], crossover, and mutation operators [8, 39], are integrated into PSO to improve its performance. Afterwards, some scholars began to pour attention into hybridizing various EAs with diverse design ideas [28, 30, 41, 42]. Recently, integrating local-searching strategies [14, 46, 47] with PSO has attracted more attentions, and the experimental results indicate that these strategies can achieve faster convergence speed and higher solution quality. Meanwhile, detecting strategies [40] and opposition-based learning strategies [38] are also introduced to help particles to jump out of locally optimal solutions.

3. SopPSO

Inspired by the ideas introduced above, we propose a sophisticated PSO algorithm (SopPSO) based on multi-level adaptation and purposeful detection. In SopPSO, adjustment strategies applying at individual and dimensional level to tune parameters are introduced to improve its comprehensive performance. Moreover, a detecting strategy and a local-searching strategy are proposed to help particles to jump out of locally optimal solutions and enhance the accuracy of solutions, respectively. The framework of SopPSO and details of its components are described as follows.

3.1. Multi-level adaptation of learning model

There are two common social learning models according to neighbor topology, i.e., global version model PSO ($GPSO$) and local version model PSO ($LPSO$), which are conductive to keeping the diversity and speeding up the convergence, respectively [12]. In SopPSO, each particle has the two candidate learning models during the evolutionary process. To offset the contradiction between the population diversity and the convergence speed, the probability that a particle chooses $LPSO$ model is much higher at the initial stage of evolution while the probability that a particle chooses the $GPSO$ is higher at the later stage of evolution. The selecting strategy of learning model is detailed as (3).

$$SL_i^t = \begin{cases} LPSO, \text{if } rand \geq (t/T)^2 \\ GPSO, \text{else} \end{cases} \tag{3}$$

where SL_i^t is social learning model adopted by the i^{th} particle in generation t; T is the maximum number of allowable generations.

However, it not means a particle need to re-select its learning model in each generation since replacing the learning model too frequently may disturb its current positive searching direction. But when and how to adjust a particle's neighbor topology and learning model?

1) When a particle choosing $LPSO$ learning model, three cases should be addressed for the above-described problems.

– The continuous generations-stagnancy $Stag_i$ of the i^{th} particle is less than a predefined individual's maximum generations-stagnancy $MaxStag_{ind}$. In this case, there is no need to change neighbor topology of the particle.

– The i^{th} particle is better than all its neighbors and $Stag_i$ is larger than $MaxStag_{ind}$. In this case, the particle may be very hard to extract beneficial information from its inferior neighbors. Thus, a reasonable strategy is random re-selecting neighbors from population for the particle.

– The i^{th} particle has more favorable neighbors than itself though $Stag_i$ is larger than $MaxStag_{ind}$. Under the condition, it may be a promising choice for the particle to re-select some specific dimensions to learn from the superior

neighbors. The motivation is that we regard there is still much useful information within the neighbors that has not been rationally utilized.

We suppose that it is beneficial for reinforcing a particle's exploration ability at the initial evolutionary state if the particle exchanges information with its exemplar on those dimensions that are far away from its corresponding dimensions. Thus, we define a variable dn_{ij}^t to indicate whether the i^{th} particle learns the j^{th} dimension from its neighbor in generation t. This operator can be interpreted as that the particle is self-confidence at the early evolutionary state, since it only learns some specific dimensions from its neighbors, and learns all dimensions from its own experience. The definition of dn_{ij}^t is described as (4).

$$dn_{ij}^t = \begin{cases} 1, \text{if } ||nb_{ij}^t - x_{ij}^t|| > r(t) \\ 0, \text{else} \end{cases} \tag{4}$$

where nb_{ij}^t is the j^{th} dimensional values of the i^{th} particle's neighbors in generation t; $r(t)$ refers to a distance threshold which linearly decreasing with the iterative generations. The updating rule of $r(t)$ is described as (5).

$$r(t+1) = r_{max} - \frac{t}{T} \cdot (r_{max} - r_{min}) \tag{5}$$

where r_{max} and r_{min} are predefined maximal and minimal distance, respectively.

Together with the aforementioned, the i^{th} particle updates its velocity based on *LPSO* model is defined in (6).

$$v_{ij}^{t+1} = w \cdot v_{ij}^t + c_1 \cdot r_1 \cdot (pb_{ij}^t - x_{ij}^t) + c_2 \cdot r_2 \cdot dn_{ij} \cdot (nb_{ij}^t - x_{ij}^t), if\ SL_i^t = LPSO \tag{6}$$

2) When a particle chooses *GPSO* as its social learning model with higher probability at the later evolutionary stage, the particle learns all dimensions from **Gb** at the social learning part but only learns some specific target dimensions from $\mathbf{Pb}_i$ at the self-cognitive part. This operator can be interpreted as that the particle realizes its own insufficiency at the later evolutionary stage, and then learns all knowledge and parts of information from **Gb** and $\mathbf{Pb}_i$, respectively. In this research, we define dp_{ij}^t described as (7) to indicate whether the i^{th} particle learns the j^{th} dimension from $\mathbf{Pb}_i$ in generation t.

$$dp_{ij}^t = \begin{cases} 1, \text{if } ||pb_{ij}^t - x_{ij}^t|| > r(t) \\ 0, \text{else} \end{cases} \tag{7}$$

As a result, the updating rule of velocity based on *GPSO* model is defined in (8).

$$v_{ij}^{t+1} = w \cdot v_{ij}^{t} + c_1 \cdot r_1 \cdot dp_{ij}^{t} \cdot (pb_{ij}^{t} - x_{ij}^{t}) + c_2 \cdot r_2 \cdot (nb_{ij}^{t} - x_{ij}^{t}), \text{if } SL_i^t = GPSO \quad (8)$$

According to the discussions above, the adaptation of learning model is detailed as Algorithm 1.

Algorithm 1. Learning_Model_Updating()

```
For each particle X_i in the population Do
  If rand ≥ (t/T)^2 Then   /* X_i chooses LPSO */
    If Stag_i ≥ MaxStag_ind Then
      If X_i is better than Nb_{i,1} and Nb_{i,2} Then
        Random reselecting two neighbors from population, i.e., N_{i,1} and N_{i,2}, for X_i;
        Updating velocity vector V_i according to Equ.(1);
      Else   /*X_i is worsen than Nb_{i,1} or Nb_{i,2} */
        Updating dn_ij^t according to Equ.(4) and (5);
        Updating velocity vector V_i according to Equ.(6);
      End If
    End If
  Else /* X_i chooses GPSO */
    Updating dp_ij^t according to Equ.(7) and (5);
    Updating velocity vector V_i according to Equ.(8);
  End If
  Update X_i according to Equ.(2);
End For
```

3.2. Detecting strategy

Although some aforementioned strategies may maintain the diversity of population and overcome premature convergence to some degree, it is unavoidable that the population may fall into a local optimum when optimizing a difficult multimodal function. To help the population jump out of a potential local optimum, a detecting strategy applied to **Gb** is proposed in this research.

To easily collect helpful information and operate the detecting operator, the searching space of each dimension of a specific problem is divided into many equal small-sized sub-regions which satisfy the following condition:

$$\bigcup_{j=1}^{Rn} s_j^i = \mathbf{S}^i \text{ and } s_j^i \bigcap s_k^i = \emptyset, j \neq k \quad (9)$$

where $\mathbf{S}^i$ is the entire searching space of the i^{th} dimension; Rn is the number of sub-regions; s_j^i and s_k^i are the j^{th} and the k^{th} sub-region ($1 \leq j, k \leq Rn$) of the i^{th} dimension, respectively.

Based on the segmentation mechanism, we can easily determine which sub-region that each component of **Gb** belongs to as well as which sub-region that **Gb** needs to detect. In this research, some historical information of the swarm is adopted to determine the advantage of each sub-region based on which **Gb** can carry out a purposeful detecting operator. For simplicity, we only record how many times that all particles' historical best positions lie within a specific sub-region after every $Cycle$ generations, which is called sampling period in this research. The advantage of each sub-region is described as (10).

$$ADV_i^j = ADV_i^j + 1, \text{if } pb_{ki} \text{ within } s_i^j \tag{10}$$

where pb_{ki} ($1 \leq i \leq D$, $1 \leq k \leq N$) is the i^{th} dimension value of the k^{th} particle's historical best position; N and D are the population size and the number of dimensions, respectively; ADV_i^j is the advantage of the sub-region s_i^j.

Accordingly to the value of ADV_i^j, the sub-regions can be categorized into three types: 1) superior sub-regions which have the maximum ADV_i^j; 2) inferior sub-regions which have the minimum ADV_i^j; and 3) moderate sub-regions which have neither the maximum nor the minimum ADV_i^j. Correspondingly, **Gb** has three detecting operators on the i^{th} dimension, the details of which are described as follows. To facilitate presentation, gb_i and adv_i denote the i^{th} dimension value of the **Gb** and ADV_i^j ($1 \leq j \leq Rn$), respectively.

– If gb_i lies within a superior sub-region, **Gb** will detect an inferior sub-region of the i^{th} dimension. The motivation behind this detecting operator is that since the swarm has never or seldom reached the inferior sub-region, and if there is a global optimum within the sub-region, it may be an efficient operator for **Gb** to detect the inferior sub-region. Even if there is no global optimum within the sub-region, the operation wastes only one evaluation.

– If gb_i belongs to an inferior sub-region, there is no requirement for **Gb** to perform the detecting operator. The reason is that a majority of individuals lying in the superior or moderate sub-regions could drag **Gb** out of the inferior sub-region after a few generations if there is no global optimum in the sub-region.

– If gb_i falls into a moderate sub-region, **Gb** will randomly select a sub-region of the i^{th} dimension to carry out the detecting operator. This progress

looks just like a random mutation operator on the i^{th} dimension.

Note that a greedy strategy is adopted in the detecting operator. More specifically, gb_i would be replaced by the detected value only if the new position improves **Gb**'s fitness. Furthermore, we use a vector $\mathbf{IMP} = \{imp_1, imp_2, ..., imp_D\}$ to record whether **Gb** has been improved or not after a series of detecting operators. Specifically, imp_i equal to 1 (or 0) indicates **Gb** is improved (or unimproved) after the detecting operator on the i^{th} dimension.

Considering that statistical information in different sampling periods may cause gb_i to detect the same sub-region many times, which making gb_i loses an opportunity to find other undetected sub-regions, we thus adopt a tabu-based detecting strategy for gb_i. For example, s_j^i will be set a tabu flag $tabu_{i,j}$ to '1' after gb_i detecting it. In this case, the subregion is no longer detected by gb_i until $tabu_{i,j}$ is reset to '0'. For a specific $\mathbf{S}^i$, while all tabu flags $tabu_{i,j}$ have been set as '1', these flags will be removed at once.

When to perform the detecting operator is another issue to be addressed. In this study, **Gb** carries out the detecting operator only when it has stagnated more than $MaxStag_{best}$, which representing the maximum generations-stagnancy of the best position that the population has achieved.

Based on the aforementioned discussions, the detecting strategy of **Gb** can be detailed as Algorithm 2.

Algorithm 2. Detecting()

TmpGb = **Gb**;
For each dimension value $tmpGb_i$ of **TmpGb Do**
  If $tmpGb_i$ within superior sub-regions
    $tmpGb_i$ is replaced by a random value within an inferior sub-region s_k^i where $tabu_{i,k}$ is 0;
  Else if $tmpGb_i$ within moderate sub-regions
    $tmpGb_i$ is replaced by a random value within a randomly selected sub-region k where $tabu_{i,k}$ is 0;
  End If
  Evaluate the new **TmpGb**;
  If TmpGb is better than **Gb Then**
    Gb=**TmpGb**; $imp_i = 1$; $tabu_{i,k} = 1$;
  Else
    $imp_i = 0$; $tabu_{i,k} = 1$;
  End If
  If all $tabu_{i,k}$ $(1 \leq k \leq Rn)$ are equal to 1 **Then**
    set each $tabu_{i,k}$ $(1 \leq k \leq Rn)$ to 0;
  End If
End For

3.3. Local-searching strategy

To improve the exploitation capability of PSO, we propose a simple local-searching strategy, based on which **Gb** carries out a series of dimension-by-dimension replacement operators after the detecting operator. For the sake of preserving favorable outcomes of the detecting operator, **Gb** only selects those dimensions, on which it has not found a more promising value after the detecting operator, to perform the replacement operator. The details of the dimension-replacement operator can be viewed in Algorithm 3.

Algorithm 3. Dimension_Replacement()

```
01: TmpGb = Gb;
02: Random selects a particle X_i from the swarm;
03: For each dimension value tmpGb_j of TmpGb Do   /*tmpGb_j is the j^th dimension of TmpGb*/
04:   If imp_j == 0 Then   /*imp_j is the j^th dimension of IMP*/
05:     tmpGb_j = x_ij; Evaluate TmpGb;
06:     If TmpGb is better than Gb Then
07:       Gb = TmpGb;
08:     End If
09:   End If
10: End For
```

3.4. Framework of SopPSO

Together with the aforementioned components, the pseudo code of SopPSO is detailed as Algorithm 4.

Algorithm 4. SopPSO()

```
01: Initialize X_i, X_i (1 ≤ i ≤ N), Rn, Cycle, MaxStag_ind, and MaxStag_best;
02: Set Stag_i=0, Stag_best=0, t=1;
03: Evaluate X_i; Initialize Pb_i = X_i, Gb = {Pb_i|min(fit(Pb_i)), 1 ≤ i ≤ N};
04: While not meet stop conditions Do
05:   Update positions of the population according to Algorithm 1;
06:   Evaluate the population;
07:   Update Pb_i, Stag_i, Gb, and Stag_best;
08:   If mod(t, Cycle)==0||Stag_best > MaxStag_best Then
09:     Carry out the detecting operator according to Algorithm 2;
10:     Carry out the local-searching operator according to Algorithm 3;
11:   End If
12:   t = t + 1; Update fes;
13: End While
14: Terminate and produce output.
```

3.5. Computational complexity

It can be seen from Algorithm 4 that the main computational complexity of SopPSO occurs in the **While**-loop (from line 04 to line 12). Obviously, the computational complexity in each loop is focus on the operators of Algorithm 1, 2, and 3 since the computational complexity of other operators in the loop

are only $O(1)$. Thus, only the computational complexities of Algorithm 1, 2, and 3 are analyzed as follows.

Algorithm 1 shows that the main difference between the multi-level adaptation of learning model in SopPSO and the classic learning model in canonical PSO is the updating rules of velocity. In SopPSO, dn_{ij}^t (or dp_{ij}^t) needs to be calculated in each generation before the updating of velocity. Since the computational complexity of the updating of dn_{ij}^t (or dp_{ij}^t) is $O(D)$ which is the same as that of velocity. So, the computational complexity of Algorithm 1 is $O(N \cdot D)$, where N and D are the population size and the total dimensions respectively. In Algorithm 2, we only carry out the detecting operator on each dimension of **Gb**, so the computational complexity of this part is $O(D)$. Similarly, the local-searching operator also only applies in each unimproved dimension of **Gb**. As a result, the computational complexity of it also is $O(D)$.

Thus, if the algorithm is stopped after a fixed number of generation T, the overall computational complexity of SopPSO is $O(T \cdot N \cdot D)$.

4. Performance of parameters and components in SopPSO

As discussed in the previous section, there are four introduced parameters (i.e., Rn, $Cycle$, $MaxStag_{ind}$, and $MaxStag_{best}$) and three components (i.e., multi-level adaptation of learning model, detecting operator, and local-searching operator) influence the performance of SopPSO. To find out how the parameters and the components impact SopPSO, 16 numerical functions, which are widely used in the literatures [14, 16, 40, 45], are selected to verify the sensitivity of the parameters and the efficiency of the components. The basic information of the benchmark functions (f_1-f_{16}) are given in Table 1.

4.1. Sensitivity analysis of each parameter

To separately analyze the effect of each parameter, four groups of experiments with Rn set to {3, 7, 10}, $MaxStag_{ind}$ set to {5, 10, 15}, $MaxStag_{best}$ set to {5, 10, 15}, and $Cycle$ set to {3, 7, 10} are carried out separately. When

Table 1: Benchmark functions f_1-f_{16}.

Name	Function	Search space	f_{min}	Properties
f_1	Sphere	$[-100, 100]^D$	0	Unimodal/Seprable
f_2	Schwefel P2.22	$[-10, 10]^D$	0	Unimodal/Seprable
f_3	Step	$[-100, 100]^D$	0	Unimodal/Seprable
f_4	Schwefel P1.2	$[-100, 100]^D$	0	Unimodal/Seprable
f_5	Diff. power	$[-100, 100]^D$	0	Unimodal/Seprable
f_6	Rosenbrock	$[-30, 30]^D$	0	Multimodal/Nonseprable
f_7	Ackley	$[-32, 32]^D$	0	Multimodal/Seprable
f_8	Alpine	$[-10, 10]^D$	0	Multimodal/Seprable
f_9	Schwefel	$[-500, 500]^D$	0	Multimodal/Seprable
f_{10}	Rastrigin	$[-5.12, 5.12]^D$	0	Multimodal/Seprable
f_{11}	Noncont_Rastrigin	$[-5.12, 5.12]^D$	0	Multimodal/Seprable
f_{12}	Griewank	$[-600, 600]^D$	0	Multimodal/Nonseprable
f_{13}	Weierstrass	$[-0.5, 0.5]^D$	0	Multimodal/Seprable
f_{14}	Generalized Penalized	$[-50, 50]^D$	0	Multimodal/Seprable
f_{15}	Schwefel P2.13	$[-\pi, \pi]^D$	0	Multimodal/Nonseprable
f_{16}	Pathological	$[-100, 100]^D$	0	Multimodal/Nonseprable

testing the influence of a particular parameter, other parameters are set as the default values. In these experiments, four default values of the parameters are all set as 10. For instance, to test the effectiveness of Rn, we use a set of values for Rn and the default values for $MaxStag_{ind}$ ($MaxStag_{ind}$=10), $MaxStag_{best}$ ($MaxStag_{best}$=10), and $Cycle$ ($Cycle$=10), respectively. The corresponding optimal values of each parameter for each problem are shown in Table 2. Moreover, the symbol "=" in Table 2 denotes there is no significant difference between the performance of the three parameter candidates of the given parameter.

The experiment on Rn indicates that the test functions are not very sensitive to Rn. There are several reasons behind the phenomenon which are explained as follows. A larger Rn causes each sub-region to have a smaller space, which enables **Gb** to take a finer detection. However, a larger Rn enables **Gb** to carry out more times of detecting operators before it finds out a promising sub-region since there are more sub-regions needed to be detected. On the contrary, a smaller Rn enables **Gb** to find out a proper but more coarse sub-region after fewer detecting operators. In other words, different Rn may cause **Gb** to have the same ability to find out a more promising result.

Table 2 shows that a smaller $Cycle$ has more favorable performance, especially in the unimodal functions (f_1, f_2, f_4, and f_5, etc.) and some multimodal functions (f_{13} and f_{16}). The reason lies behind the observable results may be that a shorter sampling period enables the swam to carry out the local-searching

Table 2: The best results of different parameters on 30D functions.

Algorithm	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
Rn	3	=	=	=	7	=	=	=	=	=	=	10	=	=	=	=
$Cycle$	3	3	=	3	3	15	=	7	15	=	=	7	3	7	15	3
$MaxStag_{ind}$	15	15	=	10	5	10	=	5	5	=	=	10	5	15	10	5
$MaxStag_{best}$	10	5	=	15	10	5	=	10	10	=	=	15	10	5	5	5

strategy (i.e., dimension-replacement operator) more frequently, which is beneficial for optimizing unimodal functions and some simple multimodal functions since there is no local optimum or complex fitness landscape misleading the searching direction of the swarm. In the course of optimizing the difficult multimodal problems, a larger $Cycle$ enables SopPSO to accumulate more helpful information to direct the population fly towards a superior region.

From the results on $MaxStag_{ind}$ and $MaxStag_{best}$, it can be observed that the performance of SopPSO with larger $MaxStag_{ind}$ and $MaxStag_{best}$ is comparable to that with smaller values of the parameters. However, this does not imply that the values of $MaxStag_{ind}$ and $MaxStag_{best}$ have no effect on the performance of SopPSO. It only means that it is necessary to further investigate how to effectively set up or tune the parameters for general problems.

Although we have a rudimentary understanding of how the new introduced parameters affect SopPSO's performance, 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, there may be dependency between the different parameters, which causes the combination of each optimal value no long to be an optimal setting for SopPSO. In the subsequent experiments, the parameters obtained after trial-and-error are set to Rn=10, $MaxStag_{ind}$=13, $MaxStag_{best}$=5, and $Cycle$=3.

4.2. Efficiency of each component

There are three strategies introduced to improve the performance of PSO: adaptive-tuning of learning model (At), local-searching strategies (Ls), and detecting (Dc). To find out how the three key strategies affect the performance of SopPSO, a series of experiments is conducted on all the 16 functions in 30 dimensions. The experimental results are listed in Table 3, in which three algorithms

named SopPSO-At, SopPSO-Ls, and SopPSO-Dc are the algorithms that removing At, Ls, and Dc from SopPSO, respectively. Note that, in SopPSO-At, the adaptive-tuning of learning model is replaced by $LPSO$ model. Furthermore, we also carry out a series of experiments to investigate how the three strategies affect the convergence speed and diversity of the population. In this study, the population deviation is denoted as (11).

$$diversity(ps) = \frac{1}{|ps| \cdot |L|} \cdot \sum_{i=1}^{|ps|} \sqrt{\sum_{j=1}^{D} (x_{ij} - \overline{x}_j)^2} \tag{11}$$

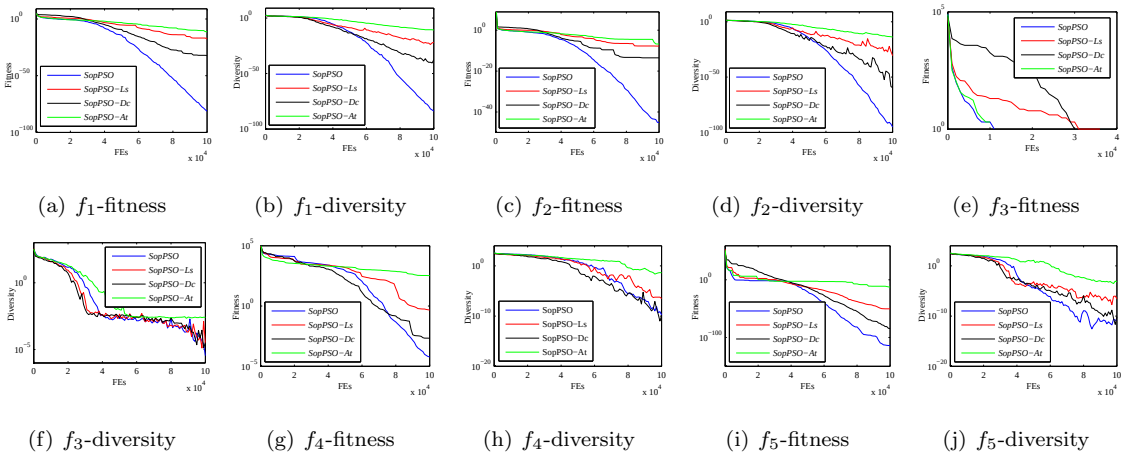
where $|ps|$ is the population size, $|L|$ is the length of the longest diagonal in the searching space, D is the dimensionality of the problem, x_{ij} is the j^{th} value of the i^{th} particle and $\overline{x}_j$ is the j^{th} value of the average point $\overline{x}$.

The comparison results of the three strategies on unimodal and multimodal functions are plotted in Fig 1 and Fig 2, respectively, in terms of convergence speed and diversity changing trend. The results show that SopPSO-At yields more promising performance on keeping the diversity of population than SopPSO not only on the 5 unimodal functions but also on the 11 multimodal functions. This property may be attributed to $LPSO$ model applied in SopPSO-At. However, it demonstrates unfavorable capability than SopPSO on all tested functions in terms of solution accuracy and convergency speed. The comparison results between SopPSO-At and SopPSO indicate that the adaptive-tuning of learning model has more promising characteristic on reaching the global optima solution with a higher convergence speed.

Comparing the results between SopPSO and SopPSO-Ls, we see that SopPSO has higher population diversity than SopPSO-Ls in the initial evolution stage on all unimodal functions and 9 out of the 11 multimodal functions except f_8 and f_{10}. On the contrary, in the later evolution stage, SopPSO rapidly loses the population diversity, and converges to a global optimal solution with a higher speed than SopPSO-Ls on the majority of the problems. In addition, SopPSO dominates SopPSO-Ls on all tested functions except f_3, in terms of the mean solutions. These phenomena validate the hypothesis that the local-searching strategy has a pleasurable performance for improving the convergence

Table 3: Testify the performance of each strategy by 30D functions.

Functions	SopPSO	SopPSO-At	SopPSO-Dc	SopPSO-Ls
f_1	6.51E-79±2.00E-78	9.42E-09±2.85E-08	2.07E-30±8.52E-30	1.72E-16±5.72E-16
f_2	8.57E-38±4.23E-37	6.45E-06±1.93E-05	1.98E-08±1.08E-07	1.49E-06±6.35E-06
f_3	0.00E+00±0.00E+00	0.00E+00±0.00E+00	5.33E-01±6.29E-01	0.00E+00±0.00E+00
f_4	8.52E-04±1.36E-03	5.15E+02±2.37E+02	1.05E-01±2.12E-01	6.43E+00±1.39E+01
f_5	1.18E-110±5.24E-110	1.22E-28±3.47E-28	6.70E-77±3.66E-76	2.58E-66±1.07E-65
f_6	7.31E+00±1.93E+01	4.66E+01±3.03E+01	3.53E+01±2.59E+01	4.86E+01±3.35E+01
f_7	3.88E-14±9.04E-15	3.26E-03±7.15E-03	2.01E-01±4.44E-01	2.69E-09±4.15E-09
f_8	2.78E-16±4.38E-16	4.06E-04±4.12E-04	2.22E-04±7.49E-04	9.57E-03±3.62E-02
f_9	2.91E-12±1.56E-12	4.21E+01±6.37E+01	3.48E+03±5.62E+03	3.95E+00±2.16E+01
f_{10}	0.00E+00±0.00E+00	2.07E-02±1.01E-02	3.18E+01±8.12E+00	4.26E-11±1.29E-10
f_{11}	0.00E+00±0.00E+00	2.42E-02±1.52E-02	3.89E+01±1.11E+01	2.06E-09±8.42E-09
f_{12}	1.95E-02±2.18E-02	3.49E-03±6.23E-03	1.36E-02±1.62E-02	7.06E-03±8.85E-03
f_{13}	5.92E-15±5.31E-15	2.46E-02±1.51E-02	3.28E-01±1.65E-01	1.21E-01±5.43E-02
f_{14}	1.66E-32±3.85E-33	6.66E-06±1.99E-05	3.46E-03±1.89E-02	2.56E-17±8.87E-17
f_{15}	1.14E+04±7.34E+03	1.66E+04±1.01E+04	2.63E+04±1.84E+04	2.54E+04±1.57E+04
f_{16}	2.35E+00±6.78E-01	2.72E+00±6.63E-01	8.87E+00±9.60E-01	3.08E+00±7.92E-01



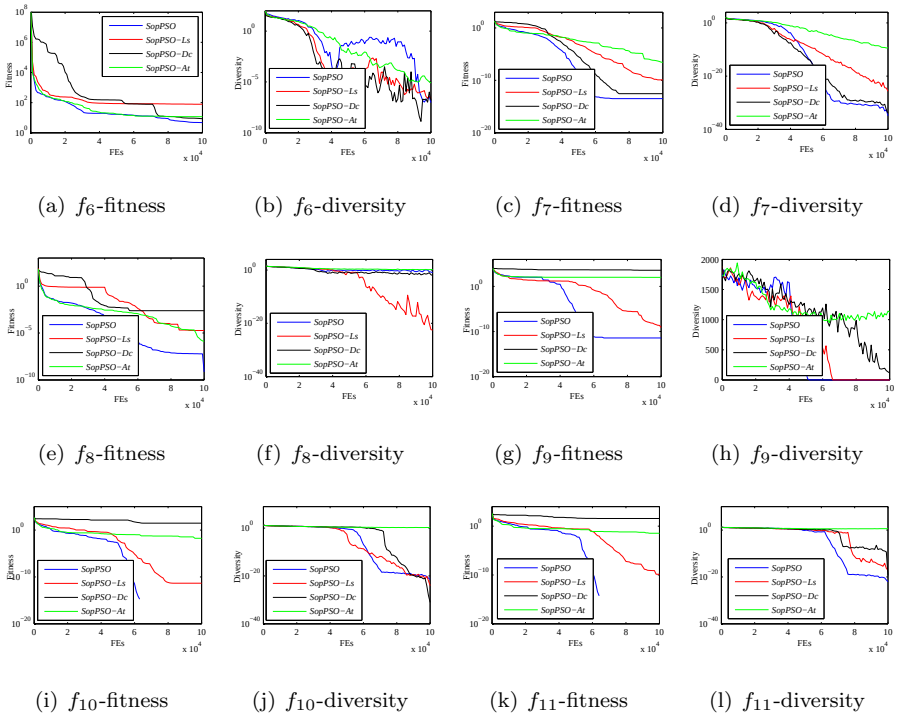
(a) f_1-fitness (b) f_1-diversity (c) f_2-fitness (d) f_2-diversity (e) f_3-fitness

(f) f_3-diversity (g) f_4-fitness (h) f_4-diversity (i) f_5-fitness (j) f_5-diversity

Figure 1: Comparisons of convergence speed and diversity of the strategies on the 5 unimodal functions.

speed of the algorithm, especially in the later evolution stage.

From the comparison, it can be observed that population diversity of SopPSO and SopPSO-Dc fall much more rapidly than that of SopPSO-Ls and SopPSO-At on the majority of the unimodal functions. Furthermore, SopPSO and SopPSO-Dc also achieve higher accurate solutions than other two algorithms on the unimodal functions. On the contrary, SopPSO-Dc yields more dreadful results on 9 out of the 11 multimodal functions in terms of solution accuracy though population diversity of it shows still fall more rapidly than that of



(a) f_6-fitness (b) f_6-diversity (c) f_7-fitness (d) f_7-diversity

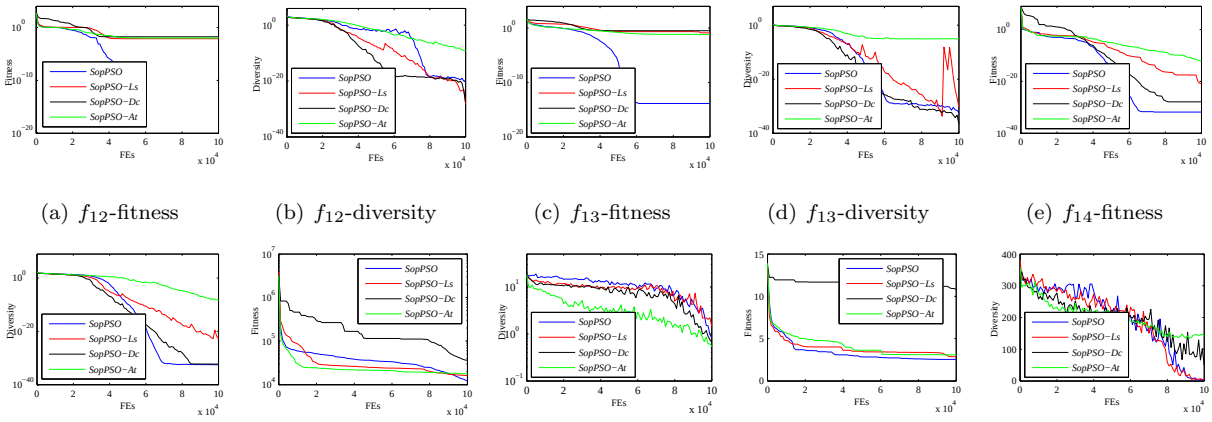
(e) f_8-fitness (f) f_8-diversity (g) f_9-fitness (h) f_9-diversity

(i) f_{10}-fitness (j) f_{10}-diversity (k) f_{11}-fitness (l) f_{11}-diversity

Figure 2: Comparisons of convergence speed and diversity of the strategies on the 11 multimodal functions.

SopPSO-Ls and SopPSO-At. Nevertheless, SopPSO not only has higher convergence speed but also achieves more accurate solutions than that of SopPSO-Dc on all the multimodal functions. The results verify that the detecting operator indeed can help a particle detect a more promising position when the swarm has been trapped into local optimum.

From the discussions above we can draw some conclusions: 1) The detecting strategy endows the algorithm with a promising performance on multimodal functions, but sacrifices its performance on unimodal functions; 2) The local-searching strategy allows the algorithm to attain competitive results on both unimodal and multimodal functions to various extents; and 3) The multi-level adaptation of learning model enables the algorithm with a pleasurable performance on different problems in varying degrees.



(a) f_{12}-fitness (b) f_{12}-diversity (c) f_{13}-fitness (d) f_{13}-diversity (e) f_{14}-fitness

(f) f_{14}-diversity (g) f_{15}-fitness (h) f_{15}-diversity (i) f_{16}-fitness (j) f_{16}-diversity

Figure 2: (Continue) Comparisons of convergence speed and diversity of the strategies on the 11 multimodal functions.

5. Comparison with other peer algorithms

5.1. Benchmark functions

To investigate the comprehensive performance of SopPSO in different environments, the CEC'13 test suite including 5 unimodal functions (F_1-F_5), 15 basic multimodal functions (F_6-F_{20}), and 8 composition functions (F_{21}-F_{28}) is adopted in this research. According to the property of the functions, we choose 1, 100, and 1000 as the accepted errors for unimodal, basic multimodal and composition functions, respectively. Due to the limited space, the definitions of the tested functions can be refereed to the literature [18].

5.2. Peer algorithms

Thirteen existing evolutionary algorithms including 9 PSO variants and 4 state-of-the-art EAs are chosen to compare with SopPSO. The configurations of each peer algorithm detailed in Table 4 are exactly the same as that used in the corresponding papers.

To fairly compare SopPSO with the other 13 algorithms, all algorithms are independently run 30 times on the tested problems in two cases, i.e., $D = 30$ and

Table 4: Thirteen peer algorithms.

Algorithm	Year	Population Topology	Parameters Settings
PSO [7]	2000	Local ring	χ=0.7298,c_1=c_2=1.49445
FDR [24]	2003	Fitness-Distance-Ration	w:[0.4,0.9],c_1=c_2=1,c_3=2
FIPS [21]	2004	Local ring	χ=0.7298,$\sum c_i$=4.1
CLPSO [17]	2006	Comprehensive Learning	w:[0.4,0.9],c=1.49445
DMSPSO [47]	2008	Dynamic Multi-swarm	χ = 0.7298,c_1=c_2=1.49445, R=10,L=100,L_FEs=200
OLPSO [46]	2011	Orthogonal Learning with Local ring	w:[0.4,0.9],c=2.0,G=5
SLPSO [14]	2012	Adaptive	w:[0.4,0.9],η=1.496,γ=0.01
CPSO-H$_k$ [3]	2004	Local ring in each subswarm	w:[0.4,0.9],c_1=c_2=1.49445
CCPSO2 [15]	2012	Local ring in each subswarm	χ=0.7298,c_1=c_2=1.49445
SaDE [25]	2009	-	Default settings in [25]
ABC [9]	2007	-	α=1
CS [44]	2009	-	α=1, P_a=0.25
TLBO [27]	2012	-	-

$D = 50$. Accordingly, the population size (N) is set to $N = D$ for all the peer algorithms except CCPSO2 and SaDE since the two algorithms have their own characteristics. In [15], there is no default values of N for $D = 30$ and $D = 50$ problems since CCPSO2 is designed for solving large-scale problems. After taking a few trial-and-error experiments, we find out that a smaller population size is beneficial for CCPSO2 on lower-dimensional functions. In this research, the population sizes of CCPSO2 are set to $N = 6$ and $N = 10$ for $D = 30$ and $D = 50$, respectively. In SaDE, the population size is set to $N = 50$ for the two cases according to the literature [25]. The maximum number of fitness evaluations ($MaxFEs$) for each run is set to $MaxFEs = 1000D$.

5.3. Comparison on solution accuracy

The comparison results, including mean value (Mean), standard deviation (Std. Dev.), minimum value (Min), and success ratio (SR) of the solutions, on the unimodal, basic multimodal and composition functions are listed in Table 5, Table 6, and Table 7, respectively. In Table 5 - Table 7, the best results of mean values on each problem among all algorithms are marked in bold. Note that t-test results between SopPSO and other peer algorithms are also demonstrated in Table 5 - Table 7. The discussion of the t-test results is detailed in subsection 5.4 and subsection 5.5.

1) *Unimodal functions*: Table 5 shows that SopPSO yields the best results on 2 out of the 5 unimodal benchmark functions in terms of the mean solution

Table 5: Comparison results on 30D unimodal functions.

Algorithm		F_1	F_2	F_3	F_4	F_5
SopPSO	Mean	4.70E-13	**1.02E-01**	1.08E+08	**1.28E-02**	1.68E-12
	Std.Dev.	7.70E-14	1.02E-01	1.11E+08	8.51E-03	1.02E-12
	Min	0.00E+00	6.69E-04	1.06E+06	2.20E-03	0.00E+00
	SR(%)	100	100	0	100	100
PSO	Mean	2.20E-13(=)	5.46E+06(+)	8.08E+08(+)	4.16E+03(+)	5.42E-13(=)
	Std.Dev.	1.47E-13	2.00E+06	6.72E+08	9.95E+02	2.13E-12
	Min	0.00E+00	1.51E+06	1.70E+08	1.47E+03	2.27E-13
	SR(%)	100	0	0	0	100
FDR	Mean	5.15E-13(=)	2.41E+05(+)	2.39E+07(-)	6.81E+02(+)	4.70E-13(-)
	Std.Dev.	1.61E-13	1.10E+05	2.23E+07	2.91E+02	7.63E-14
	Min	2.27E-13	5.19E+04	3.13E+03	2.26E+02	3.41E-13
	SR(%)	100	0	0	0	100
FIPS	Mean	**0.00E+00**(-)	2.99E+05(+)	4.72E+10(+)	4.06E+04(+)	**0.00E+00**(-)
	Std.Dev.	0.00E+00	1.48E+05	4.36E+10	1.05E+04	0.00E+00
	Min	0.00E+00	8.29E+04	4.84E+09	1.88E+04	0.00E+00
	SR(%)	100	0	0	0	100
CLPSO	Mean	1.21E-13(=)	1.59E+07(+)	3.82E+07(-)	9.97E+03(+)	1.14E-13(=)
	Std.Dev.	1.13E-13	2.50E+06	1.74E+07	1.42E+03	0.00E+00
	Min	0.00E+00	1.06E+07	8.39E+06	6.90E+03	1.14E-13
	SR(%)	100	0	0	0	100
DMSPSO	Mean	6.06E-14(-)	1.59E-01(=)	4.05E+07(=)	7.60E-01(+)	1.02E-13(=)
	Std.Dev.	8.89E-14	2.63E-01	3.44E+07	1.35E+00	2.63E-14
	Min	0.00E+00	7.45E-04	7.60E+03	2.04E-03	1.14E-13
	SR(%)	100	96.7	0	96.7	100
OLPSO	Mean	6.06E-14(-)	4.40E+07(+)	3.04E+10(+)	1.35E+05(+)	1.29E-13(=)
	Std.Dev.	8.89E-14	1.57E+07	1.07E+10	1.83E+04	2.63E-14
	Min	0.00E+00	1.29E+07	1.25E+10	9.62E+04	1.14E-13
	SR(%)	100	0	0	0	100
SLPSO	Mean	5.61E-13(+)	2.85E+06(+)	3.36E+09(+)	4.84E+04(+)	9.80E-12(=)
	Std.Dev.	1.13E-13	1.30E+06	2.81E+09	6.75E+03	5.80E-12
	Min	4.55E-13	7.77E+05	9.93E+07	3.13E+04	1.93E-12
	SR(%)	100	0	0	0	100
CPSO-H$_k$	Mean	9.03E+02(+)	7.74E+06(+)	1.21E+10(+)	1.24E+04(+)	7.00E+01(+)
	Std.Dev.	5.81E+02	5.86E+06	9.03E+09	7.55E+03	7.29E+01
	Min	4.55E-13	2.94E+05	1.56E+08	1.87E+03	3.41E-13
	SR(%)	16.7	0	0	0	46.7
CCPSO2	Mean	2.96E-13(=)	1.80E+07(+)	3.09E+09(+)	1.48E+05(+)	3.60E-13(=)
	Std.Dev.	1.27E-13	7.12E+06	2.37E+09	3.15E+04	1.34E-13
	Min	0.00E+00	7.88E+06	3.83E+08	8.84E+04	1.14E-13
	SR(%)	100	0	0	0	100
SaDE	Mean	**0.00E+00**(-)	7.00E+05(+)	1.13E+08(=)	5.11E+03(+)	**0.00E+00**(-)
	Std.Dev.	0.00E+00	2.69E+05	7.10E+07	1.94E+03	0.00E+00
	Min	0.00E+00	1.74E+05	5.35E+05	1.53E+03	0.00E+00
	SR(%)	100	0	0	0	100
ABC	Mean	1.37E-11(+)	5.52E+08(+)	1.65E+11(+)	1.66E+05(+)	7.30E-03(+)
	Std.Dev.	9.32E-12	9.41E+07	3.60E+10	3.33E+04	3.23E-03
	Min	1.14E-12	3.02E+08	6.08E+10	9.40E+04	2.16E-03
	SR(%)	100	0	0	0	100
CS	Mean	1.88E+02(+)	9.08E+02(+)	**1.05E+03**(-)	8.10E+02(+)	3.49E-13(=)
	Std.Dev.	2.89E+02	5.95E+02	7.61E+02	5.63E+02	2.71E-13
	Min	0.00E+00	4.52E+01	1.03E+02	1.46E+01	1.14E-13
	SR(%)	73.3	0	0	0	100
TLBO	Mean	1.76E-11(+)	6.03E+05(+)	3.99E+08(=)	1.48E+03(+)	4.82E-10(=)
	Std.Dev.	1.30E-11	2.81E+05	5.95E+08	7.41E+02	8.78E-10
	Min	2.50E-12	1.72E+05	1.42E+06	1.84E+02	2.39E-12
	SR(%)	100	0	0	0	100

accuracy. On the contrary, FIPS and SaDE not only achieve the best results on other 2 unimodal functions (i.e. F_1 and F_5) but also obtain the global optimal solutions on them. The reason why SopPSO can not yield the best results on F_1 and F_5 as FIPS and SaDE is that the detecting operator in SopPSO wastes a

Table 6: Comparison results on 30D basic multimodal functions.

Algorithm		F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}
SopPSO	Mean	4.90E-10	3.95E+01	**2.09E+01**	2.40E+01	**1.02E-01**	**2.33E-14**	1.10E+02	1.55E+02
	Std.Dev.	7.24E-10	9.07E+00	5.46E-02	3.10E+00	6.62E-02	2.52E-14	4.61E+01	2.36E+01
	Min	5.91E-12	1.53E+01	2.08E+01	1.82E+01	7.40E-03	0.00E+00	3.08E+01	1.00E+02
	SR(%)	100	100	100	100	100	100	53.3	0
PSO	Mean	4.86E+01(+)	3.73E+01(=)	2.10E+01(=)	**1.75E+01**(-)	2.77E+01(+)	4.12E+01(+)	1.01E+02(=)	1.47E+02(=)
	Std.Dev.	3.73E+01	1.02E+01	6.06E-02	1.14E+00	2.19E+01	7.15E+00	2.33E+01	1.79E+01
	Min	1.64E+01	1.64E+01	2.08E+01	1.53E+01	8.84E-02	2.48E+01	5.08E+01	1.09E+02
	SR(%)	100	100	100	100	100	100	50.0	0
FDR	Mean	1.32E+01(+)	**2.25E+01**(-)	**2.09E+01**(=)	1.98E+01(-)	1.30E-01(=)	2.62E+01(+)	5.72E+01(-)	1.15E+02(-)
	Std.Dev.	5.69E+00	6.44E+00	5.54E-02	3.33E+00	3.45E-02	4.82E+00	1.53E+01	2.46E+01
	Min	2.55E-01	4.57E+00	2.08E+01	1.30E+01	2.22E-02	1.49E+01	3.38E+01	5.98E+01
	SR(%)	100	100	100	100	100	100	96.7	36.7
FIPS	Mean	2.25E+01(+)	4.48E+02(+)	2.10E+01(=)	1.84E+01(-)	1.70E-01(+)	6.95E+01(+)	6.02E+01(-)	**1.06E+02**(-)
	Std.Dev.	1.04E+01	2.53E+02	4.25E-02	2.59E+00	5.28E-02	3.03E+01	2.03E+01	3.09E+01
	Min	9.62E+00	1.68E+02	2.09E+01	9.57E+00	5.42E-02	1.59E+01	1.99E+01	5.20E+01
	SR(%)	100	100	100	100	100	90.0	90.0	50.0
CLPSO	Mean	2.29E+01(+)	4.91E+01(+)	**2.09E+01**(=)	2.61E+01(+)	3.71E-01(+)	3.32E-02(=)	8.41E+01(-)	1.23E+02(-)
	Std.Dev.	7.02E+00	8.25E+00	4.08E-02	1.72E+00	1.55E-01	6.41E-02	1.16E+01	1.44E+01
	Min	1.64E+01	2.77E+01	2.08E+01	2.19E+01	8.874E-02	0.00E+00	5.17E+01	8.34E+01
	SR(%)	100	100	100	100	100	100	83.3	16.7
DMSPSO	Mean	**5.64E-11**(-)	2.27E+01(-)	**2.09E+01**(=)	1.99E+01(=)	1.11E-01(=)	5.64E+01(+)	**4.96E+01**(-)	9.32E+01(-)
	Std.Dev.	3.10E-11	1.06E+01	4.08E-02	3.38E+00	4.15E-02	1.27E+01	1.04E+01	1.81E+01
	Min	3.07E-12	3.09E+00	2.08E+01	1.31E+01	2.22E-02	2.69E+01	2.69E+01	4.93E+01
	SR(%)	100	100	100	100	100	100	100	50.0
OLPSO	Mean	4.89E+01(+)	1.71E+02(+)	2.10E+01(=)	2.40E+01(=)	1.61E+02(+)	2.47E+00(+)	1.45E+02(+)	2.11E+02(+)
	Std.Dev.	3.38E+00	3.63E+01	2.97E-02	1.92E+00	5.96E+01	3.42E+00	5.19E+01	2.65E+01
	Min	4.32E+01	9.80E+01	2.08E+01	2.01E+01	1.79E+01	0.00E+00	6.86E+01	1.55E+02
	SR(%)	100	3.3	100	100	26.7	100	26.7	0
SLPSO	Mean	2.77E+01(+)	9.65E+01(+)	2.10E+01(=)	3.07E+01(+)	3.77E-01(+)	1.33E-13(=)	1.08E+02(=)	1.70E+02(=)
	Std.Dev.	1.80E+01	1.53E+01	6.07E-02	1.30E+00	1.30E-01	3.44E-14	2.45E+01	2.74E+01
	Min	7.43E+00	6.19E+01	2.09E+01	1.18E+01	1.18E-01	5.68E-14	5.77E+01	1.11E+02
	SR(%)	100	73.3	100	100	100	100	50.0	0.0
CPSO-H_k	Mean	9.45E+01(+)	1.54E+02(+)	**2.09E+01**(=)	2.96E+01(+)	2.81E+02(+)	4.25E+01(+)	2.10E+02(+)	2.84E+02(+)
	Std.Dev.	2.47E+01	3.18E+01	4.96E-02	2.67E+00	1.12E+02	1.69E+01	3.67E+01	3.51E+01
	Min	3.39E+01	7.83E+01	2.08E+01	2.43E+01	9.05E+01	1.09E+01	1.02E+02	2.11E+02
	SR(%)	56.7	10.0	100	100	3.3	100	0	0
CCPSO2	Mean	4.94E+01(+)	1.64E+02(+)	2.12E+01(=)	3.27E+01(+)	7.45E+00(+)	2.03E-13(=)	1.88E+02(+)	2.89E+02(+)
	Std.Dev.	1.15E+01	2.86E+01	8.91E-02	4.02E+00	9.07E+00	7.77E-14	4.99E+01	5.54E+01
	Min	1.70E+01	1.07E+02	2.10E+01	2.26E+01	2.81E-01	5.68E-14	7.68E+01	1.87E+02
	SR(%)	96.7	0	100	100	100	100	6.7	0
SaDE	Mean	3.38E+01(+)	5.02E+01(+)	**2.09E+01**(=)	2.57E+01(=)	3.52E-01(+)	5.47E+00(+)	6.58E+01(-)	1.29E+02(-)
	Std.Dev.	2.38E+01	1.54E+01	5.95E-02	2.69E+00	1.30E-01	4.15E+00	1.14E+01	2.49E+01
	Min	5.28E-01	5.28E-01	2.09E+01	1.40E+01	1.06E-01	9.95E-01	3.98E+01	7.59E+01
	SR(%)	100	100	100	100	100	100	100	16.7
ABC	Mean	2.57E+01(+)	4.31E+02(+)	2.10E+01(=)	3.96E+01(+)	3.07E+01(+)	2.09E+02(+)	2.38E+02(+)	2.41E+02(+)
	Std.Dev.	3.77E-01	4.12E+01	2.96E-02	1.10E+00	1.86E+01	9.89E+00	9.27E+00	9.84E+00
	Min	2.45E+01	2.45E+01	2.08E+01	3.69E+01	4.21E+00	1.86E+02	2.06E+02	2.14E+02
	SR(%)	100	0	100	100	100	0	0	0
CS	Mean	1.28E+01(+)	4.72E+02(+)	**2.09E+01**(=)	2.96E+01(+)	1.41E-02(-)	4.33E+01(+)	1.57E+02(+)	1.83E+02(+)
	Std.Dev.	2.64E+00	4.85E+02	4.75E-02	1.26E+00	1.08E-02	7.16E+00	2.37E+01	2.40E+01
	Min	4.92E+00	4.79E+01	2.08E+01	2.47E+01	8.12E-07	2.44E+01	1.01E+02	1.04E+02
	SR(%)	100	10	100	100	100	100	10.0	10.0
TLBO	Mean	4.49E+01(+)	9.34E+01(+)	**2.09E+01**(=)	3.04E+01(+)	1.28E+00(=)	1.99E+02(+)	1.56E+02(+)	2.47E+02(+)
	Std.Dev.	2.43E+01	2.25E+01	5.95E-02	2.26E+00	1.99E+00	3.50E+01	3.83E+01	4.11E+01
	Min	1.53E+01	4.80E+01	2.09E+01	2.30E+01	4.68E-02	1.11E+02	5.86E+01	1.69E+02
	SR(%)	100	60.0	100	100	100	0	10.0	0

few of fitness evaluations since there is no local optimum within such problems.

2) *Basic multimodal functions*: It can be seen from Table 6 that SopPSO has very competitive performance on multimodal functions since it achieves the best results on 7 out of the 15 multimodal functions. The result validates

Table 6: (Continue) Comparison results on 30D basic multimodal functions.

Algorithm		F_{14}	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}	F_{20}
SopPSO	Mean	**5.97E-02**	4.61E+03	1.33E+00	**3.04E+01**	1.50E+02	**8.18E-01**	**1.01E+01**
	Std.Dev.	2.52E-02	4.28E+02	2.70E-01	8.43E-03	4.97E+01	2.27E-01	1.00E+00
	Min	5.46E-12	3.49E+03	8.32E-01	3.04E+01	6.66E+01	2.89E-01	8.69E+00
	SR(%)	100	0	100	100	30.0	100	100
PSO	Mean	1.58E+03(+)	4.44E+03(=)	2.51E+00(+)	6.95E+01(+)	2.15E+02(+)	3.68E+00(+)	1.23E+01(+)
	Std.Dev.	2.66E+02	5.88E+02	1.94E-01	7.27E+00	1.89E+01	8.22E-01	4.91E-01
	Min	8.86E+02	2.64E+03	1.68E+00	5.20E+01	1.66E+02	2.45E+00	1.09E+01
	SR(%)	0	0	100	100	0	100	100
FDR	Mean	1.20E+03(+)	3.85E+03(-)	2.10E+00(+)	7.34E+01(+)	1.52E+02(=)	3.18E+00(+)	1.45E+01(+)
	Std.Dev.	2.25E+02	5.56E+02	2.98E-01	8.76E+00	4.62E+01	6.14E-01	8.73E-01
	Min	7.12E+02	1.68E+03	1.47E+00	4.73E+01	6.33E+01	1.50E+00	9.30E+00
	SR(%)	0	0	100	96.7	26.7	100	100
FIPS	Mean	1.18E+03(+)	4.06E+03(=)	2.48E+00(+)	4.99E+01(+)	1.77E+02(+)	1.32E+04(=)	1.48E+01(+)
	Std.Dev.	2.31E+02	1.90E+03	1.61E-01	1.14E+01	5.77E+00	2.34E+04	2.79E-01
	Min	3.65E+02	1.14E+03	2.05E+00	3.60E+01	1.61E+02	1.76E+00	1.29E+01
	SR(%)	0	0	100	93.3	0	70.0	100
CLPSO	Mean	4.49E+01(+)	4.82E+03(=)	2.27E+00(+)	4.67E+01(+)	2.01E+02(+)	3.31E+00(+)	1.31E+01(+)
	Std.Dev.	1.84E+01	3.16E+02	2.41E-01	1.90E+00	7.90E+00	3.07E-01	4.24E-01
	Min	1.70E+01	4.05E+03	1.48E+00	4.17E+01	1.79E+02	2.78E+00	1.23E+01
	SR(%)	93.3	0	100	100	0	100	100
DMSPSO	Mean	2.28E+03(+)	**3.03E+03**(-)	1.08E+00(-)	7.56E+01(+)	7.44E+01(-)	3.52E+00(+)	1.46E+01(+)
	Std.Dev.	3.36E+02	2.55E+02	2.37E-01	7.54E+00	9.48E+00	6.22E-01	7.28E-01
	Min	1.43E+03	2.12E+03	4.37E-01	4.95E+01	5.63E+01	1.95E+00	1.09E-01
	SR(%)	0	0	100	100	96.7	100	100
OLPSO	Mean	8.56E+01(+)	7.32E+03(+)	3.50E+00(+)	3.10E+01(+)	2.36E+02(+)	2.69E+00(+)	1.44E+01(+)
	Std.Dev.	6.72E+01	9.21E+02	3.21E-01	2.28E-01	1.51E+01	4.03E-01	5.09E-01
	Min	1.04E-01	4.53E+03	2.75E+00	3.07E+01	1.83E+02	1.64E+00	1.29E+01
	SR(%)	66.7	0	100	100	0	100	100
SLPSO	Mean	6.35E-02(=)	4.25E+03(-)	**1.05E+00**(-)	**3.04E+01**(=)	1.44E+02(=)	9.85E-01(+)	1.19E+01(+)
	Std.Dev.	3.05E-02	5.14E+02	2.55E-01	8.46E-03	2.17E+01	1.89E-01	7.62E-01
	Min	1.54E-02	2.61E+03	5.44E-01	3.04E+01	9.29E+01	5.13E-01	1.00E+01
	SR(%)	100	0	100	100	3.3	100	100
CPSO-H$_k$	Mean	1.41E+03(+)	4.55E+03(=)	9.85E-01(-)	3.19E+01(+)	1.69E+02(=)	3.74E+01(+)	1.46E+01(+)
	Std.Dev.	3.70E+02	5.17E+02	3.32E-01	7.16E-01	3.85E+01	5.92E+01	4.58E-01
	Min	6.31E+02	3.47E+03	4.00E-01	3.10E+01	1.09E+02	1.13E+00	1.27E+01
	SR(%)	0	0	100	100	0	96.7	100
CCPSO2	Mean	1.34E+01(=)	4.73E+03(=)	2.48E+00(+)	2.95E+01(=)	1.87E+02(+)	1.06E+00(+)	1.48E+01(+)
	Std.Dev.	2.21E+01	5.48E+02	6.93E-01	1.96E+00	4.34E+01	2.43E-01	2.81E-01
	Min	3.10E-02	3.28E+03	1.18E+00	5.69E-02	1.12E+02	3.52E-01	1.30E+01
	SR(%)	96.7	0	100	100	0	100	100
SaDE	Mean	1.00E+01(+)	3.38E+03(-)	2.16E+00(+)	3.09E+01(+)	**7.18E+01**(-)	3.46E+00(+)	1.02E+01(=)
	Std.Dev.	8.49E+00	4.18E+02	1.77E-01	2.70E-01	8.97E+00	9.75E-01	6.71E-01
	Min	8.33E-02	2.18E+03	1.71E+00	3.05E+01	5.01E+01	1.51E+00	8.70E+00
	SR(%)	96.7	0	100	100	100	100	100
ABC	Mean	8.09E+03(+)	8.12E+03(+)	2.45E+00(+)	2.54E+02(+)	2.63E+02(+)	7.62E+03(+)	1.50E+01(+)
	Std.Dev.	1.74E+02	2.63E+02	2.49E-01	1.07E+01	8.91E+00	2.89E+03	6.53E-06
	Min	7.58E+03	7.50E+03	1.09E+00	2.12E+02	2.27E+02	1.97E+03	1.50E+01
	SR(%)	0	0	100	0	0	0	100
CS	Mean	2.37E+03(+)	4.25E+03(-)	1.68E+00(+)	1.53E+02(+)	2.17E+02(+)	1.04E+03(+)	1.26E+01(+)
	Std.Dev.	2.44E+02	2.56E+02	2.57E-01	1.10E+01	1.79E+01	6.54E+02	3.30E-01
	Min	1.83E+03	3.43E+03	1.15E+00	1.17E+02	1.70E+02	4.24E+00	1.16E+01
	SR(%)	0	0	100	0	0	13.3	100
TLBO	Mean	2.98E+03(+)	6.94E+03(+)	2.44E+00(+)	1.95E+02(+)	2.74E+02(+)	6.04E+01(+)	1.24E+01(+)
	Std.Dev.	5.56E+02	2.93E+02	2.24E-01	3.29E+01	3.65E+01	2.86E+01	6.26E-01
	Min	1.79E+03	5.56E+03	1.76E+00	1.35E+02	2.08E+02	1.62E+01	1.11E+01
	SR(%)	0	0	100	0	0	86.7	100

the hypothesis that the detecting operator in SopPSO can help **Gb** jump out of potential local optimum, and then gives SopPSO a better explorative capability. Furthermore, DMSPSO also shows a promising characteristic on the multimodal functions. In this context, the purposeful detecting operator and the multi-

Table 7: Comparison results on 30D composition functions.

Algorithm		F_{21}	F_{22}	F_{23}	F_{24}	F_{25}	F_{26}	F_{27}	F_{28}
SopPSO	Mean	3.32E+02	**8.76E+01**	5.88E+03	2.66E+02	**2.46E+02**	**2.00E+02**	9.50E+02	**2.87E+02**
	Std.Dev.	8.12E+01	3.42E+01	9.93E+02	9.55E+00	6.50E+01	3.79E-03	5.94E+01	2.49E+01
	Min	1.00E+02	1.93E-12	3.90E+03	2.52E+02	2.52E+02	2.00E+02	7.82E+02	1.00E+02
	SR(%)	100	100	0	100	100	100	70.0	100
PSO	Mean	2.89E+02(=)	1.74E+03(+)	4.43E+03(-)	2.61E+02(=)	2.84E+02(+)	2.66E+02(+)	8.20E+02(-)	8.09E+02(+)
	Std.Dev.	7.14E+01	3.52E+02	5.34E+02	4.77E+00	7.07E+00	7.40E+01	4.85E+01	3.73E+02
	Min	2.00E+02	7.64E+02	3.22E+03	2.47E+02	2.70E+02	2.00E+02	7.03E+02	3.00E+02
	SR(%)	100	6.7	0	100	100	100	100	53.3
FDR	Mean	3.07E+02(=)	1.32E+03(+)	4.13E+03(-)	2.47E+02(-)	2.81E+02(=)	3.21E+02(+)	8.07E+02(-)	4.52E+02(=)
	Std.Dev.	6.37E+01	2.76E+02	6.76E+02	1.61E+01	7.01E+00	4.02E+01	6.89E+01	3.33E+02
	Min	1.00E+02	5.79E+02	2.61E+03	2.19E+02	2.66E+02	2.00E+02	5.83E+02	1.00E+02
	SR(%)	100	16.7	0	100	100	100	100	83.3
FIPS	Mean	3.60E+02(=)	1.62E+03(+)	4.82E+03(-)	2.77E+02(+)	2.86E+02(+)	3.15E+03(+)	9.01E+02(-)	8.64E+02(+)
	Std.Dev.	8.90E+01	4.41E+02	1.55E+03	9.12E+00	7.28E+00	5.63E+01	7.61E+01	6.77E+02
	Min	2.00E+02	6.13E+02	2.25E+03	2.57E+02	2.69E+02	2.00E+02	6.83E+02	3.00E+02
	SR(%)	100	100	0	100	100	100	83.3	60.0
CLPSO	Mean	2.77E+02(-)	2.10E+02(+)	5.39E+03(-)	2.61E+02(=)	2.85E+02(+)	2.07E+02(=)	8.33E+02(-)	3.00E+02(=)
	Std.Dev.	3.45E+01	6.55E+01	3.24E+02	8.37E+00	4.02E+00	1.03E+01	2.10E+02	2.37E-13
	Min	2.00E+02	1.31E+02	4.11E+03	2.44E+02	2.73E+02	2.01E+02	4.08E+02	3.00E+02
	SR(%)	100	100	0	100	100	100	83.3	100
DMSPSO	Mean	3.28E+02(=)	2.19E+03(+)	**3.48E+03**(-)	**2.39E+02**(-)	2.84E+02(+)	2.56E+02(+)	**7.72E+02**(-)	3.86E+02(+)
	Std.Dev.	7.70E+01	5.41E+02	4.75E+02	1.55E+01	1.07E+01	6.72E+01	9.91E+01	2.09E+02
	Min	1.00E+02	9.05E+02	2.28E+03	2.11E+02	2.12E+02	2.00E+02	5.07E+02	1.00E+02
	SR(%)	100	3.3	0	100	100	100	96.7	90.0
OLPSO	Mean	2.23E+02(-)	3.73E+02(+)	7.55E+03(+)	2.74E+02(+)	2.85E+02(+)	2.34E+02(+)	9.54E+02(=)	1.43E+03(+)
	Std.Dev.	3.97E+01	1.56E+02	1.13E+03	4.16E+00	4.53E+00	4.35E+01	4.22E+01	2.03E+02
	Min	2.00E+02	1.39E+02	5.19E+03	2.64E+02	2.02E+02	2.02E+02	8.42E+02	8.14E+02
	SR(%)	100	96.7	0	100	100	100	83.3	6.7
SLPSO	Mean	3.22E+02(=)	1.15E+02(=)	4.90E+03(-)	2.74E+02(+)	2.90E+02(+)	2.36E+02(+)	1.06E+03(+)	3.55E+02(+)
	Std.Dev.	6.49E+01	4.61E+01	5.28E+02	7.06E+00	7.77E+00	5.65E+01	6.89E+01	1.02E+02
	Min	2.00E+02	3.38E-05	3.02E+03	2.72E+02	2.72E+02	2.00E+02	7.77E+02	3.00E+02
	SR(%)	100	100	0	100	100	100	16.7	96.7
CPSO-H$_k$	Mean	4.94E+02(+)	1.65E+03(+)	5.73E+03(=)	2.87E+02(+)	3.03E+02(+)	3.47E+02(+)	1.13E+03(+)	1.63E+03(+)
	Std.Dev.	1.82E+02	4.45E+02	5.27E+02	7.44E+00	7.96E+00	4.88E+01	5.51E+01	3.79E+02
	Min	1.00E+02	4.08E+02	4.60E+03	2.63E+02	2.82E+02	2.00E+02	9.23E+02	3.00E+02
	SR(%)	96.7	10.0	0	100	100	100	3.3	13.3
CCPSO2	Mean	3.02E+02(=)	1.14E+02(=)	5.99E+03(=)	2.97E+02(+)	3.08E+02(+)	3.56E+02(+)	1.16E+03(+)	3.79E+02(=)
	Std.Dev.	7.55E+01	3.33E+01	7.74E+02	8.63E+00	7.89E+00	4.18E+01	1.02E+02	1.42E+02
	Min	2.00E+02	8.65E-01	4.04E+03	2.61E+02	2.88E+02	2.00E+02	9.70E+02	3.00E+02
	SR(%)	100	100	0	100	100	100	10.0	96.7
SaDE	Mean	3.47E+02(=)	1.17E+02(=)	3.72E+03(-)	2.40E+02(-)	2.78E+02(=)	2.09E+02(+)	9.68E+02(=)	2.93E+02(=)
	Std.Dev.	7.81E+01	1.38E+01	4.10E+02	6.36E+00	6.54E+00	1.65E+01	6.96E+01	1.29E+01
	Min	2.00E+02	1.93E+01	2.75E+03	2.25E+02	2.52E+02	2.00E+02	5.81E+02	1.00E+02
	SR(%)	100	100	0	100	100	100	100	100
ABC	Mean	2.86E+02(-)	8.34E+03(+)	8.50E+03(+)	3.02E+02(+)	3.01E+02(+)	3.72E+02(+)	1.33E+03(+)	3.00E+02(=)
	Std.Dev.	4.02E+01	2.46E+02	1.96E+02	2.62E+00	2.52E+00	2.75E+01	1.94E+01	6.26E-03
	Min	2.00E+02	6.70E+03	7.70E+03	2.92E+02	2.95E+02	2.95E+02	1.28E+03	3.00E+02
	SR(%)	100	0	0	100	100	100	0	100
CS	Mean	**1.95E+02**(-)	2.96E+03(+)	5.07E+03(-)	2.86E+02(+)	3.00E+02(+)	**2.00E+02**(+)	1.07E+03(+)	3.00E+02(=)
	Std.Dev.	2.68E+01	3.06E+02	3.86E+02	4.86E+00	4.88E+00	4.57E-02	1.04E+02	4.19E-09
	Min	1.01E+02	2.01E+03	3.71E+03	2.71E+02	2.88E+02	2.00E+02	4.02E+02	3.00E+02
	SR(%)	100	0	0	100	100	100	13.3	100
TLBO	Mean	3.25E+02(=)	2.83E+03(+)	6.58E+03(+)	2.77E+02(+)	2.93E+02(+)	2.89E+02(+)	1.01E+03(+)	2.07E+03(+)
	Std.Dev.	6.31E+01	5.57E+02	7.94E+02	5.85E+00	7.08E+00	8.33E+01	6.05E+01	5.04E+02
	Min	2.00E+02	1.59E+03	2.35E+03	2.52E+02	2.79E+02	2.00E+02	8.97E+02	3.00E+02
	SR(%)	100	0	0	100	100	100	50.0	3.3

swarm may be the favorable strategies to handle multimodal problems.

3) *Composition functions*: Table 7 shows that SopPSO also yields the best comprehensive performance in terms of the solution accuracy since it achieves the best results on 4 out of the 8 composition functions. Note that DMSPSO

also shows the most favorable property on 3 composition functions. Although SaDE does not obtain the best result on any composition functions, it yields the best reliability among the functions since it achieves the 100 percent success ratio among 7 out of the 8 composition functions.

It can be observed from Table 5 - Table 7 that SopPSO yields the best results on 13 out of the 28 benchmark functions in terms of the mean solution accuracy, followed by DMSPSO, SaDE, and CS. Furthermore, SopPSO shows the best reliability among the 28 functions since the average success ratio it achieves among the test functions is 80.4%, followed by DMSPSO, SaDE, and CLPSO, the average success ratios of which are 79.6%, 79%, and 74.2%, respectively.

5.4. Comparison on statistical results

According to the acceptable error given for each problem in this study, we present the comparison of the statistical results of the 14 algorithms. The second row in Table 8 indicates the number of functions that are solved (#S), partially solved (#PS), and never solved (#NS) by the peer algorithms in 30D and 50D cases. A function is called solved, partially solved, or never solved if an algorithm reaches the corresponding acceptable error level over all 30 runs, over some of (but not all) 30 runs, or over none of 30 runs, respectively. The third row in Table 8 demonstrates the average success ratio of the 14 algorithms on the 28 benchmark functions in 30D and 50D cases. The fourth row in Table 8 indicates the number of functions that SopPSO is significantly better than (#+), significantly worse then (#-), and almost the same as (#=) the compared algorithms in 30D and 50D cases in terms of t-test results, respectively.

Table 8: Statistic results between SopPSO and the 13 peer algorithms.

	D	SopPSO	PSO	FDR	FIPS	CLPSO	DMSPSO	OLPSO	SLPSO	CPSO-H$_k$	CCPSO2	SaDE	ABC	CS	TLBO
#S,#PS,#NS	30	21,3,4	21,3,4	16,6,6	14,7,7	18,4,6	17,7,4	14,7,7	17,5,6	9,10,9	15,5,8	21,2,5	13,0,15	13,6,9	12,5,11
	50	20,1,7	12,5,11	14,4,10	13,4,11	15,5,8	14,4,10	16,2,10	16,3,9	9,7,12	14,4,10	18,4,6	8,2,18	12,2,14	10,5,13
Average SR (%)	30	80.5	64.6	69.9	62.4	74.2	79.6	61.1	69.3	44.8	64.5	79.0	46.4	50.4	50.4
	50	73.6	53.0	55.8	55.0	61.2	58.6	59.8	62.4	43.2	59.8	66.4	31.4	44.3	47.9
#+,#-,#=	30	-	16,3,9	12,9,7	15,9,4	14,6,8	11,11,6	22,2,4	15,3,10	23,1,4	17,0,11	11,10,7	25,1,2	21,4,3	23,0,5
	50		20,4,4	16,6,6	22,5,1	17,2,9	13,9,6	20,3,5	14,0,14	24,3,1	23,1,4	11,11,6	27,0,1	22,3,3	21,1,6

From the second row in Table 8, it can be seen that SopPSO has the best

performance among the 14 algorithms on the two dimensional cases since it achieves the best results on both 30D and 50D cases in terms of the number of solved functions. SaDE yields slightly worse results than SopPSO. Furthermore, CLPSO also shows more reliable property since the number of problems solved by it is 18 which is slightly smaller than that of SaDE.

The results listed in the fourth row in Table 8 show that SopPSO outperforms the other 12 algorithms on 30D case in terms of the t-test results except achieves the same performance as DMSPSO. On 50D case, SopPSO outperforms the other 12 algorithms in terms of the t-test results except achieves the same performance as SaDE. The number of problems that SopPSO significantly better than the other algorithms is much larger than that SopPSO performs significantly worse than the other algorithms. The comparison results verify that SopPSO has more reliable property on higher dimension problems.

Comparison results between SopPSO and other 13 peer algorithms indicate that SopPSO outperforms the other 12 algorithms in terms of the t-test results except DMSPSO, which yields the almost the same results as SopPSO.

5.5. Comparison on nonparametric tests

Generally, it is necessary to use nonparametric tests to analyze the experimental results. In this research, two-tailed t-test and Friedman-test, are carried out to validate the performance of all the compared algorithms. The results of t-test on 30D problems are presented in Table 5 - Table 7. The symbols "+", "-", and "=" denote that SopPSO is significantly better than, significantly worse than, and almost the same as the corresponding competitor algorithm, respectively. The summary statistic of t-test results on 30D and 50D cases are provided in the second row of Table 8, in which the symbol "#+", "#-", and "#=" denote the number of the functions on which the performance of SopPSO is significantly better than, significantly worse than, and almost the same as the corresponding competitor algorithm, respectively.

The results of Friedman-test on 30D and 50D benchmark functions are listed in Table 9 and Table 10, respectively. In addition, we also separately carry out

Friedman-test on the unimodal, basic multimodal, and composition functions, the results of which are also listed in the tables, in which each algorithm and its rankings are listed in ascending order (the lower the better).

It can be seen from Table 9 that SopPSO offers the best overall performance on the 28 functions in 30D case, while SaDE is the second best, followed by DMSPSO and FDR. From the Friedman-test results on the 5 unimodal functions we can see that DMSPSO and SaDE yield more favorable performance than SopPSO. The reason of it may be that the detecting operator in SopPSO wastes a few of fitness evaluations since there is no local optimum within such unimodal problems. On the contrary, SopPSO offers the best performance on the 15 multimodal functions. The results verify the effectiveness of the detecting operator on multimodal functions. On the composition functions, SopPSO yields the second best performance which is slightly worse than SaDE. Table 10 shows that SopPSO also achieves the most favorable performance on both unimodal and multimodal problems in terms of the average ranking, followed by SaDE, FDR, and DMSPSO. The results verify that SopPSO yields a reliable comprehensive performance on different cases.

Table 9: Friedman-test of all compared algorithms on 30D functions.

Average	Algorithm	Ranking	Unimodal: F_1-F_5		Multimodal: F_6-F_{20}		Composition: F_{21}-F_{28}	
Rank			Algorithm	Ranking	Algorithm	Ranking	Algorithm	Ranking
1	SopPSO	4.25	DMSPSO	2.90	SopPSO	3.77	SaDE	3.63
2	SaDE	4.70	SaDE	4.60	SaDE	4.90	SopPSO	4.81
3	DMSPSO	4.71	SopPSO	5.00	DMSPSO	4.93	CLPSO	4.88
4	FDR	5.20	FDR	5.20	FDR	4.97	DMSPSO	5.56
5	CLPSO	5.91	CS	5.40	SLPSO	5.67	FDR	5.75
6	SLPSO	6.80	CLPSO	6.20	CLPSO	6.37	PSO	6.25
7	PSO	7.36	FIPS	6.20	FIPS	7.80	SLPSO	6.94
8	CS	7.57	PSO	7.60	PSO	7.93	CS	7.44
9	FIPS	7.77	TLBO	8.40	CS	8.37	OLPSO	7.50
10	OLPSO	8.68	OLPSO	9.10	CPSO-H$_k$	9.10	FIPS	8.81
11	CCPSO2	9.50	CCPSO2	9.60	OLPSO	9.23	CCPSO2	9.88
12	TLBO	10.00	SLPSO	10.00	CCPSO2	9.27	TLBO	10.56
13	CPSO-H$_k$	10.30	CPSO-H$_k$	11.60	TLBO	10.23	ABC	11.25
14	ABC	12.25	ABC	13.20	ABC	12.47	CPSO-H$_k$	11.75
Statistic	123.860		32.111		69.881		39.741	
p value	0.000		0.002		0.000		0.000	

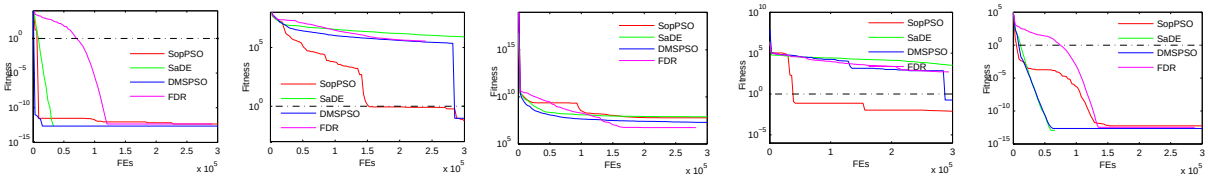
5.6. Comparison on convergence speed

Due to the limitation of space, only the results on 30D case are included in this paper. In all benchmark functions, we only choose SopPSO to compare

Table 10: Friedman-test of all compared algorithms on 50D functions.

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	SopPSO	3.98	SaDE	3.30	SopPSO	3.23	SaDE	3.50
2	SaDE	4.41	DMSPSO	3.40	FDR	5.10	SopPSO	5.13
3	FDR	5.13	SopPSO	4.40	DMSPSO	5.40	PSO	5.13
4	DMSPSO	5.64	FDR	5.00	SaDE	5.27	OLPSO	5.25
5	CLPSO	6.79	FIPS	5.90	SLPSO	5.60	FDR	5.75
6	FIPS	6.95	CLPSO	6.40	FIPS	7.13	CLPSO	6.00
7	SLPSO	7.04	CS	6.80	CLPSO	7.33	FIPS	7.25
8	PSO	7.34	TLBO	7.40	PSO	8.17	DMSPSO	7.50
9	OLPSO	7.93	PSO	8.40	CCPSO2	8.33	CS	8.00
10	CS	8.43	SLPSO	9.00	CPSO-H$_k$	8.43	SLPSO	8.50
11	CCPSO2	8.98	OLPSO	9.60	OLPSO	8.80	CCPSO2	8.94
12	CPSO-H$_k$	9.61	CPSO-H$_k$	10.80	CS	9.20	TLBO	10.50
13	TLBO	9.86	CCPSO2	11.00	TLBO	10.33	CPSO-H$_k$	11.06
14	ABC	12.93	ABC	13.60	ABC	12.67	ABC	13.0
Statistic	120.912		34.987		69.525		43.478	
p value	0.000		0.001		0.000		0.000	

the convergence speed on 30D functions with SaDE, DMSPSO, and FDR since the three algorithms have more favorable performance in terms of the Friedman-test results. The convergence progresses of the peer algorithms on the unimodal, basic multimodal and composition functions are plotted in Fig. 3, Fig. 4, and Fig. 5, respectively. The black dash-dotted lines in the figures are the predefined acceptable errors.

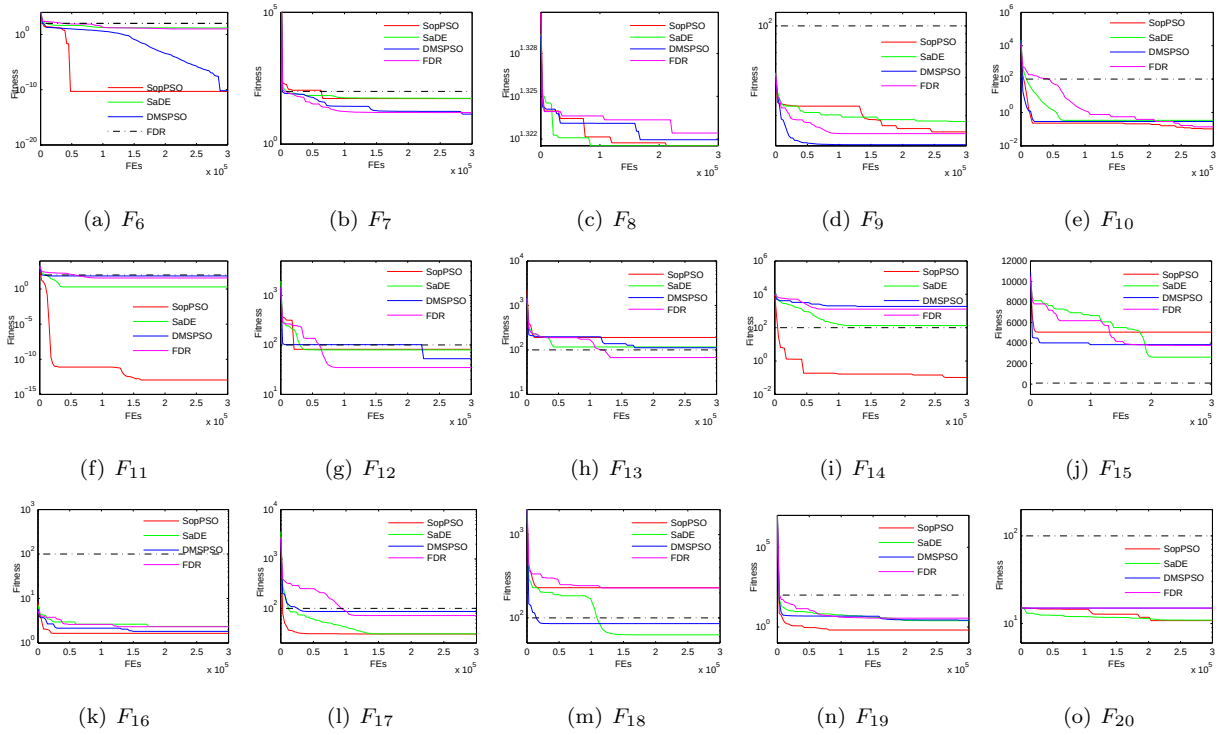


(a) F_1 (b) F_2 (c) F_3 (d) F_4 (e) F_5

Figure 3: Convergence progress of SopPSO and other three outstanding algorithms on 30D unimodal functions.

It can be observed from Fig. 3 that SopPSO yields the fastest convergence speed on 2 unimodal functions (i.e., F_2 and F_4) as well as achieves the highest accuracy on them. Furthermore, SopPSO and DMSPSO fulfill the predefined acceptable errors with a higher convergence speed on F_1 and F_5 than other algorithms.

The results demonstrated in Fig. 4 show SopPSO obtains the fastest conver-



(a) F_6 (b) F_7 (c) F_8 (d) F_9 (e) F_{10}

(f) F_{11} (g) F_{12} (h) F_{13} (i) F_{14} (j) F_{15}

(k) F_{16} (l) F_{17} (m) F_{18} (n) F_{19} (o) F_{20}

Figure 4: Convergence progress of SopPSO and other three outstanding algorithms on 30D basic multimodal functions.

gence speed on 6 out the 15 basic multimodal problems (i.e., F_6, F_{11}, F_{14}, F_{16}, F_{17}, and F_{19}) on which it also yields more favorable performance in terms of solutions accuracy. In addition, DMSPSO shows faster and stable convergence speed on F_6, F_7, F_8, and F_9. The experimental results verify that the multi-level adaptation of learning model and detecting operator in SopPSO as well as the sub-swarm mechanism in DMSPSO are in favor of the optimization of multimodal functions. It is worth to note that SopPSO has the faster convergence speed on F_{12} and F_{15} at the initial evolutionary stage though it not yields the promising performance on the functions in terms of solution accuracy.

Although all the algorithms show almost the same performance on F_8 in terms of the solution accuracy, SopPSO displays the fastest convergence speed

on it. The reason of it is possibly that the detecting strategy helps swarm to find out a promising subregion, and the local-searching strategy enables the swarm to carry out a more refined searching within the small-sized subregion.

It can be seen from Fig. 5 that SopPSO not only yields the best solution accuracy but also has the highest convergency speed on F_{22}, F_{26}, and F_{28}. On F_{25}, SopPSO shows the best performance in terms of the solution accuracy with a steady decline convergency process.

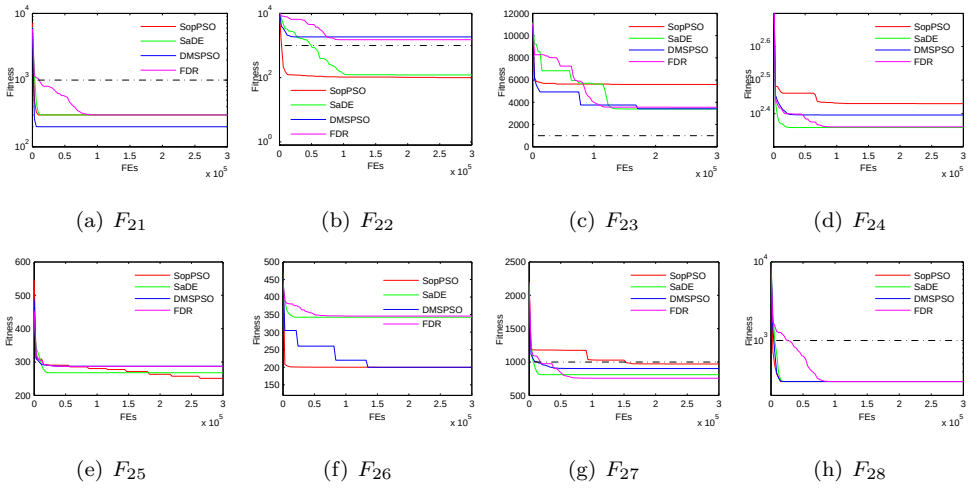
From the convergence progress, it can be seen that SopPSO has been trapped into some local optimal solutions on F_{21}, F_{23}, and F_{24} too early, and the detecting strategy is unable to drag the population out of the local optimal positions. Although SopPSO does not yield the highest accuracy on F_{21} and F_{23}, it has a higher convergency speed at the initial stage of evolution. The results verify that the local-searching operator in SopPSO does have a positive effect on accelerating the convergence while the detecting operator is beneficial for optimizing difficult problems to some extent though the performance of it needs to be further improved.

The detailed discussion of the comparison on 50D functions is not included in this paper since it has the similar characteristics as that on 30D functions.

6. Conclusion

This paper presents a variant of PSO named as a sophisticated PSO (SopPSO). During the evolutionary process, a multi-level adaptation of learning model, including neighbor topology adjustment and target dimensions adjustment, is proposed to accommodate a particle to the current fitness landscape and then to carry out an efficient search. In addition, the information of past experience is collected to help the population take a purposeful detecting operator, and then drags it out of a possible local optimum. At last, a simple local-searching strategy is proposed to accelerate the convergence speed of SopPSO.

From the comparison results, several conclusions can be drawn for SopPSO: 1) The multi-level adaptation of learning model applied not only among individ-



(a) F_{21} (b) F_{22} (c) F_{23} (d) F_{24}

(e) F_{25} (f) F_{26} (g) F_{27} (h) F_{28}

Figure 5: Convergence progress of SopPSO and other three outstanding algorithms on 30D composition functions.

ual level but also among dimensional level enables a particle to perform a more suitable searching process according to its fitness landscape; 2) The statistical information extracted from past experience can help the swarm carry out an efficient detecting operator and then enables it to jump out of a local optimum; 3) The simple local-searching strategy also displays some positive performance on optimization of complex problems; and 4) SopPSO shows more promising property on higher dimension problems in terms of the comparison results with other peer algorithms.

Although the detecting module obtains a competitive characteristic testified by the extensive experiments, it does not show pleasurable performance on multimodal non-separable functions. Our future works will focus on how to find the interactions among different variables for the non-separable problems based on historical knowledge extracted from optimization process, and then help the population carry out more efficient detecting operators for difficult problems.

Acknowledgment. This study was funded by the National Natural Science Foundation of China (Grant No.: 61663009, 61602174, 61562028), the National

Natural Science Foundation of Jiangxi Province (Grant No.: 20161BAB202064, 20161BAB212052, 20151BAB207022), and the National Natural Science Foundation of Jiangxi Provincial Department of Education (Grant No.: GJJ150539, GJJ150496).

7. Reference

[1] P.J. Angeline, Using selection to improve particle swarm optimization, in: Proc. of Cong. on Evolutionary Computation, 1998, pp. 84-89.

[2] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition, in: Proc. of Cong. on Evolutionary Computation, 2007, pp. 4661-667.

[3] V.D. Bergh, A.P. Engelbrecht, A cooperative approach to particle swarm optimization, IEEE Trans. Evol. Comput. 8(3)(2004) 225-239.

[4] S.I. Birbil, S.C. Fang, An electromagnetism-like mechanism for global optimization, J. Global Optim. 25(2003) 263-282.

[5] A. Chatterjee, P. Siarry, Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization, Comput. Oper. Res. 33(2004) 859-871.

[6] G. Ciuprina, D. Ioan, I. Munteanu, Use of intelligent-particle swarm optimization in electormagnetics, IEEE Trans. Magn. 38(2)(2002) 1037-1040.

[7] R. Eberhart, Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, in: Proc. of Cong. on Evolutionary Computation, 2000, pp. 84-88.

[8] H. Higashi, H. Iba, Particle swarm optimization with Gaussian mutation, in: Proc. of the IEEE Swarm Intelligence Symposium, 2003, pp. 72-79.

[9] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) Algorithm, J. Global Optim. 39(3)(2007) 459-171.

[10] J. Kennedy, Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance, in: Proc. of Cong. on Evolutionary Computation, 1999, pp. 1931-1938.

[11] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proc. of the IEEE Int. Conf. on Neural Network, 1995, pp. 1942-1948.

[12] J. Kennedy, R. Mendes, Population structure and particle swarm performance. in: Proc. of Cong. on Evolutionary Computation, 2002, pp. 1671-1676.

[13] K. Kentzoglanakis, M. Poole, Particle swarm optimization with an oscillating inertia weight, in: Proc. of the IEEE Conf. on Genetic and Evolutionary Computation, 2009, pp. 1749-1750.

[14] C. Li, S. Yang, T.T. Nguyen, A self-learning particle swarm optimizer for global optimization problems, IEEE Trans. Syst. Man Cybernet. Part B: Cybernet. 42(3)(2012) 627-646.

[15] X.D. Li, X. Yao, Cooperatively coevolving particle swarms for large scale optimization, IEEE Trans. Evol. Comput. 16(2)(2012) 210-224.

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

[17] J.J. Liang, A.K. Qin, P.N. Suganthan, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10(3)(2006) 281-295.

[18] J.J. Liang, B.Y. Qu, P.N. Suganthan, Alfredo G. Hernández-Díaz, Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization, Technical Report, http://www.ntu.edu.sg/home/epnsugan/, 2013.

[19] S.H. Ling, H.H.C. Iu, K.Y. Chan, Hybrid particle swarm optimization with wavelet mutation and its industrial applications, IEEE Trans. Syst. Man Cybernet. Part B: Cybernet. 38(3)(2008) 743-763.

[20] A. Marco, d.O. Montes, S. Thomas, Frankenstein's PSO: A composite particle swarm optimization algorithm, IEEE Trans. Evol. Comput. 13(5)(2009) 1120-1132.

[21] R. Mendes, J. Kennedy , J. Neves, The fully informed particle swarm: simpler,maybe better, IEEE Trans. Evol. Comput. 8(3)(2004) 204-210.

[22] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95(2016): 51-67.

[23] V.K. Patel, V.J. Savsani, Hear transfer search (HTS): A novel optimization algorithm, Inform. Sci. 324(2015): 217-246.

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

[25] A.K. Qin, V.L. Huang, Suganthan P.N., Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13(2)(2009) 398-417.

[26] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput. 8(3)(2004) 240-255.

[27] R.V. Rao, V.J. Savsani, and D.P. Vakharia, Teaching-Learning-Based optimization: An optimization method for continuous non-linear large scale problems, Inform. Sci. 183(1)(2012):1-15.

[28] S. Sayah, A. Hamouda, A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems, Appl. Soft Comput. 13(2013) 1608-1619.

[29] P. Savsani, V. Savsani, Passing vehicle search (PVS): A novel metaheuristic algorithm, Appl. Math. Model. 40(5-6)(2015) 3951-3978.

[30] A. Sedki, D. Ouazar, Hybrid particle swarm optimization and differential evolution for optimal design of water distribution systems, Adv. Eng. Inform. 26(2012) 582-591.

[31] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: Proc. of the IEEE World Cong. of Computational Intelligence, 1998, pp.68-73.

[32] Y. Shi, R.C. Eberhart, Fuzzy adaptive particle swarm optimization, in: Proc. of Cong. on Evolutionary Computation, 2001, pp.101-106.

[33] H. Soh, Y.S. Ong, Q.C. Nguyen, Q.H. Nguyen, Discovering unique, low-energy pure water isomers: Memetic exploration, optimization and landscape analysis, IEEE Trans. Evol. Comput. 14(3)(2010) 419-437.

[34] Pandian M. Vasant, Meta-Heuristics optimization algorithms in engineering, business, economics, and finance, IGI Global Publishing Hershey, 2013.

[35] Pandian M. Vasant, Handbook of research on novel soft computing intelligent algorithms: Theory and practical applications, IGI Global Publishing Hershey, 2014.

[36] Pandian. Vasant, Handbook of research on artificial intelligence techniques and algorithms, IGI Global Publishing Hershey, 2015.

[37] M.P. Wachowiak, R. Smolikova, Y.F. Zheng, An approach to multimodal biomedical image registration utilizing particle swarm optimization, IEEE Trans. Evol. Comput. 8(3)(2004) 280-288.

[38] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, M. Ventresca, Enhancing particle swarm optimization using generalized opposition-based learning, Inform. Sci. 181(2011) 4699-4714.

[39] C. Wei, Z. He, Y. Zhang, Swarm directions embedded in fast evolutionary programming, in: Proc. of Cong. on Evolutionary Computation, 2002, pp.1278-1283.

[40] 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. 23(2014) 76-90.

[41] B. Xin, J. Chen, Z.H. Peng, An adaptive hybrid optimizer based on particle swarm and differential evolution for global optimization, Sci. China Inform. Sci. 53(5)(2010) 980-989.

[42] B. Xin, J. Chen, J. Zhang, H. Fang, Z.H. Peng, Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: A review and taxionomy, IEEE Trans. Syst. Man Cybernet. Part C: Appl. Rev. 42(5)(2012) 744-767.

[43] G. Xu, An adaptive parameter tuning of particle swarm optimiztion algorithm, Appl. Math. Comput. 219(2009) 4560-4569.

[44] X.S. Yang, S. Deb, Cuckoo search via Lévy flights, in: Proc. of World Cong. on Nature & Biologically Inspired Computing, 2009, pp. 210-214.

[45] Z.H. Zhan, J. Zhang, Y. Li, Adaptive particle swarm optimization, IEEE Trans. Syst. Man Cybernet. Part B: Cybernet. 39(6)(2009) 1362-1381.

[46] Z.H. Zhan, Y. Li, Y.H. Shi, Orthogonal learning particle swarm optimization, IEEE Trans. Evol. Comput. 15(6)(2011) 832-847.

[47] S.Z. Zhao, J.J. Liang, P.N. Suganthan, Dynamic multi-swarm particle optimizer with local search for large scale global optimization, in: Proc. of Cong. on Evolutionary Computation, 2008, pp.3845-3852.

[48] Y.J. Zheng, H.F. Ling, Q. Guan, Adaptive parameters for a modified comprehensive learning particle swarm optimizer, Math. Probl. Eng. 68(2009) 939-955.