A fitness-based multi-role Particle swarm optimization

Xuewen Xia*[a,b], Ying Xing[a,b], Bo Wei[a,b], Yinglong Zhang[a,b], Xiong Li[a,b], Xianli Deng[a,b], Ling Gui[c]

[a] *School of Software, East China Jiaotong University, Nanchang 330013, China*
[b] *Intelligent Optimization and Information Processing Lab., East China Jiaotong University, Nanchang 330013, China*
[c] *School of Economics and Management, East China Jiaotong University, Nanchang 330013, China*

Abstract

In the canonical particle swarm optimization (PSO), "self-cognitive" and "social-learning" play important roles in searching for promising solutions since the two parts can help particles efficiently share their valuable information. In this paper, a fitness-based multi-role PSO (FMPSO) is proposed, in which a new component, named as "subsocial-learning" part, is added to particles' velocity update rule. Moreover, a fitness-based multi-role PSO is proposed, in which three roles, i.e., leader, rambler, and follower, are assigned for particles at each generation based on their fitness. Accordingly, different learning weights for the three parts are introduced for different roles, the aim of which is to help the population carry out various search mechanisms. Furthermore, a particle can adjust its learning exemplar and objective dimensions for the three different parts according the particle's role. During the evolutionary process, two tuning operators are introduce to adjust particles' roles and objective dimensions. The dynamical and diverse learning models of the particles enable the population to deal with different situations. Moreover, a local searching operator based on BFGS Quasi-Newton method is proposed to refine the best solution at the later evolutionary stage. Experimental results show that FMPSO outperforms other state-of-art PSOs on a majority of functions in CEC2005 and CEC2013 test suites in terms of the global search ability, solution accuracy, and convergence speed. Two statistical tests also verify the promising performance of FMPSO. Furthermore, experiments aiming to analysis the sensitivity and effectiveness of

the new proposed components in FMPSO are also conducted.

Keywords: Particle swarm optimization, Fitness-based multi-role, Periodical adjustment, Local-searching

---

1. Introduction

Particle swarm optimization (PSO) proposed by Kennedy and Eberhart in 1995 [1, 2] is a popular swarm intelligence algorithm. During the past decades, PSO has achieved great success on many real applications, such as mechanical design [3], shop scheduling [4], complex network clustering [5], and image segmentation [6]. Extensive studies reveal that PSO's capability of finding a global optimum mainly depends upon its two characteristics [7, 8, 9, 10]: explorative and exploitative capability. However, similar to other population based algorithm, PSO also suffers from the contradiction between the explorative and exploitative capability [11]. For example, explorative capability is beneficial for multimodal functions, but it sacrifices the convergence speed on unimodal functions. On the contrary, exploitative capability plays a more important role on unimodal functions than multimodal functions. It is an intuitive idea that enhancing PSO's explorative (or exploitative) capabilities for a specific problem. Nevertheless, it is unrealistic to improve PSO's searching features in advance since it is very hard, if not impossible, to obtain essential characteristics of a given problem.

Recently, various improvements, such as time-varying parameters [11, 12, 13, 14], dynamic neighbor topology [15, 16], multi-swarm techniques [17, 18], and hybrid strategy [19, 20, 21], are proposed for PSO to achieve a trade-off between explorative capability and exploitative capability, and then enhance its comprehensive performance. In this research, inspired by these researches, we propose a fitness-based multi-role particle swarm optimization (FMPSO) in which some new features are introduced to improve its comprehensive performance. Firstly, the update rule of velocity is expended from 3 parts in the canonical PSO to 4 parts including inertia part, self-cognition part, social-learning part, and

subsocial-learning part. Secondly, particles are assigned different roles based on their fitness. Accordingly, particles that playing different roles will be assigned different learning weights for the three parts, i.e., self-cognition, social-learning, and subsocial-learning. Moreover, the objective dimensions that particles learning from the three parts are also depending on their roles. The aim of the multi-role is to help the population carry out various search mechanisms. Thirdly, two tuning operators called "global tuning" and "local tuning" are introduce to adjust particles' learning model during the evolutionary process. Lastly, a local searching operator based on BFGS Quasi-Newton method is proposed to refine the best solution at the later evolutionary stage.

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 FMPSO algorithm are described in Section 3. Extensive experiments between FMPSO and other existing state-of-the-art algorithms on the CEC2005 test suite are introduced in Section 4. Moreover, sensitivity analysis of components involved in FMPSO are also detailed in this section. Finally, conclusions are given in Section 5.

2. Particle Swarm Optimization

2.1. Canonical PSO

In particle swarm optimization (PSO), each individual is deemed as a volumeless particle, and the flight trajectory of population is treated as a continuous optimization process. During the search process in a hyperspace, the ith particle is associated with two vectors, i.e., 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 the searching space. The vector $\mathbf{X}_i$ is regarded as a candidate solution to the problem while the vector $\mathbf{V}_i$ is treated as the searching direction and step size of the ith particle. At each generation, a particle updates its velocity and position relying on two vectors: the particle's personal historical best position $\mathbf{Pb}_i = [pb_{i1}, pb_{i1}, ..., pb_{iD}]$ and its neighbor's best-so-far position

$\mathbf{Nb}_i = [nb_{i1}, nb_{i1}, ..., nb_{iD}]$. The updating rules of $\mathbf{V}_i$ and $\mathbf{X}_i$ are defined as (1) and (2), respectively.

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

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

where w represents the inertia weight determining how much the previous velocity is preserved; c_1 and c_2 are known as two acceleration coefficients determining the relative learning weights for $\mathbf{Pb}_i$ and $\mathbf{Nb}_i$, which 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 position and velocity of the ith particle on the jth dimension, respectively, at generation t. Without loss of generality, minimization problems are considered in this paper.

2.2. PSO variants

Since proposed in 1995, PSO has attracted much attention from specialists in different areas. Various mechanisms are introduced to improve the performance of PSO during the last few decades. According to the different objectives to be dealt with, these modifications can be generally categorized into three cases: time-varying parameters, updating learning mechanism, multi-swarm technique, and hybrid strategy.

1) *Time-varying parameters.* Considering targets in different evolutionary stages are various, for example, the goals of early stage and later stage are global searching capability and local searching ability, respectively, many researchers propose time-varying parameters to achieve a trade-off between the explorative and the exploitative ability of PSO [22, 23]. The most ubiquitous update rules of w, c_1 and c_2 are introduced in [24, 25]. Since different problems have their own characteristics and different evolutionary stages have various targets to be dealt with, PSO desires for many adaptive capabilities to tune its parameters. For example, Zhan [11] proposed an adaptive strategy to adjust w, c_1 and c_2 based on an evolutionary state estimation (ESE) mechanism relying on the information

of population distribution and particles' fitness. In [13], a self-adaptive method for adjusting w is proposed to follow a given nonlinear ideal average velocity by feedback control. Many experiments demonstrate the very reliable performance of the adaptive regulations of parameters, both in unimodal and multimodal problems [11, 13, 26, 27].

2) *Updating learning mechanism.* Since the neighbor topology determines a particle's learning mechanism, it has attracted much attention of researchers. Although many studies suggest that a swarm with fewer connections is beneficial for multimodal problems while more connections is more appropriate for unimodal problems [7, 22, 28], 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. So, quite a few dynamic adjustments for the topology are proposed in recent years aiming to enrich particles' learning mechanism. In [15, 29], connections of the topology are increased (or decreased) along with the generations in order to meet the targets of different evolutionary stages. Apart from these generation-dependent update rules of neighbor topology, many tuning strategies for learning mechanism associated with other features are proposed for PSO [30, 31, 32, 33, 34].

To endow population with more intelligence to deal with different complex situations, a self-learning mechanism was proposed in [16]. In the self-learning particle swarm optimizer (SLPSO), each particle has a set of four roles relying on its local fitness landscape. Accordingly, four different learning mechanisms enable the particle to independently deal with different situations. The experiments demonstrate the effectiveness of the adaptive learning framework both on global search capability and overcoming premature phenomena to some extent.

3) *Multi-swarm technique.* Multi-swarm techniques, including niching methods, are introduced to EAs to allow maintenance of a population's diverse which is very helpful for multimodal optimization problems. Recently, some variants extending the single population PSO to the multi-swarm scheme have also become a branch of the research hotpots [17, 18, 33]. In [17, 18], the entire population is divided into many sub-groups carrying out the search process in parallel.

The lower diffusion speed of information among the small-sized sub-groups ensures the algorithms' population diversity. Moreover, a dynamic regrouping operator accelerates the convergence speed of the population, and then improving the accuracy of solutions. In [33], an entire population is divided into two subpopulations which choose different learning mechanisms aiming to focus on explorative and exploitation in parallel.

In addition, a multi-swarm PSO was proposed in [35], in which a mixed search behavior and a cooperative mechanism are introduced to keep the population diversity and avoid premature convergence. The different learning mechanisms in various sub-swarms enable the population has a very lower the probability of getting trapped into local optimum [35, 36, 37].

4) *Hybrid strategy.* Considering that different intelligence algorithms and operators have their own merits, many researchers pour much attention on integrating various algorithms or/and operators to handle complex problems. For example, three genetic operators, i.e., selection operator [38], crossover operator [39] and mutation operator [40], are integrated into PSO to improve its performance. Furthermore, some scholars began to pour much attention into designing more promising algorithms by integrating various algorithms. For example, numerous hybrids of differential evolution (DE) and PSO have emerged from many researchers with diverse design ideas [21, 41]. Recently, hybridizing various local search strategies [16, 18, 42, 43, 44] with PSO has attracted more attention, and the experimental results indicate that these PSO variants achieve the favorable trade-off between the convergence speed and the population's diversity.

3. Fitness-based Multi-role PSO

In SLPSO [16], each particle can select a proper operator from a set of four operator candidates according to its selection ratios which are undated based on the particle's relative performance during different evolutionary stages. Accordingly, each particle playing various roles in different generations can indepen-

dently deal with different situations according to the current fitness landscape.

Meanwhile, we know that there are two main neighbor topologies of the PSO algorithm (see (1)), called *gbest* (global best) and *lbest* (local best). Extensive experiments indicate that the two topology models have different performance on different problems. Specifically, the *gbest* model has a faster convergence while the *lbest* has stronger global search capability.

Inspired by the improvements, we propose a modified PSO algorithm called fitness-based multi-role PSO (FMPSO) in this research. The main characteristics of FMPSO can be summarized as follows.

1) In FMPSO, the update rule of velocity is expended from 3 parts in the canonical PSO to 4 parts including inertia part, self-cognitive part, social-learning part, and subsocial learning part. The inertia part in FMPSO is the same as that in the canonical PSO. In FMPSO, a particle's learning exemplars of the self-cognitive part, the social-learning part, and the subsocial-learning part are the best positions achieved by the particle itself, the entire population, and the particle's neighbors, respectively. Furthermore, when updating the velocity, a particle does not learn all dimensions from one exemplar. Specifically, the particle learn different dimensions from different exemplars according to its learning weight from the exemplars. Consequently, the update rule of velocity can be detailed as (3).

$$\begin{aligned} v_{ij}^{t+1} = w \cdot v_{ij}^{t} + c_1^i \cdot r_1 \cdot s_{d1}^i \cdot (pb_{ij}^t - x_{ij}^t) + c_2^i \cdot r_2 \cdot s_{d2}^i \cdot (gb_j^t - x_{ij}^t) \\ + c_3^i \cdot r_3 \cdot s_{d3}^i \cdot (lb_{ij}^t - x_{ij}^t) \end{aligned} \tag{3}$$

where w is the inertia weight; r_1, r_2, and r_3 are three random numbers generated in the interval [0, 1] uniformly; pb_{ij}^t, gb_j^t, and lb_{ij}^t represent the values of the jth dimension of $\mathbf{Pb}_i$, $\mathbf{Gb}$, and $\mathbf{Lb}_i$ at generation t, respectively; c_1^i, c_2^i and c_3^i are there acceleration coefficients determining the i particle's relative learning weights for $\mathbf{Pb}_i$, $\mathbf{Gb}$, and $\mathbf{Lb}_i$, respectively; s_{d1}^i, s_{d2}^i, and s_{d3}^i are three objective-dimension vectors representing which dimensions the ith particle learns from the corresponding exemplars. Specifically, '1' or '0' in the objective-dimension vec-

tors indicate that the particle selects or not selects the corresponding dimension from an exemplar as its learning objective, respectively. For example, s^i_{d1}=[1, 0, 0, 1, 1] for a problem with 5 variables indicates that the ith particle will learn from $\mathbf{Pb}_i$ on the first, the fourth, and the fifth dimensions.

2) Based on the fitness, individuals in FMPSO can be divided into three categories, i.e., elite, ordinary, and subordinate. Accordingly, the particles that belong to the three categories can be assigned three different roles, i.e., leader, rambler, and follower, respectively. If a particle is a leader, which mean the particle belongs to the elite class, it will pay more attention on self-cognitive part. In this case, the acceleration coefficients meet constraints $c^i_1 > c^i_2 > c^i_3$ and $\parallel s^i_{d1} \parallel > \parallel s^i_{d2} \parallel > \parallel s^i_{d3} \parallel$, where $\parallel X \parallel$ represents the number of '1' in the vector X. The operator can be explained that the leader is more self-confidence. When a particle is an ordinary particle, it will play a rambler role which means that the particle will learn much more information from its local neighbors, i.e., the sub-social part. In this condition, three acceleration coefficients should meet constraints $c^i_3 > c^i_2 > c^i_1$ and $\parallel s^i_{d3} \parallel > \parallel s^i_{d2} \parallel > \parallel s^i_{d1} \parallel$. The operator can be regarded as that the rambler is sensible individual. While a particle is a follower, which mean the particle belongs to the subordinate class, it will strength the learning intensity on social-learning. In this case, the individual will extract much more information from $\mathbf{Gb}$ as well as $\mathbf{Lb}_i$. This operator can be considered as that follower is very self-contemptuous since it only pay little attention on its historical experience. In this case, three acceleration coefficients should meet constraints $c^i_2 > c^i_3 > c^i_1$ and $\parallel s^i_{d2} \parallel > \parallel s^i_{d3} \parallel > \parallel s^i_{d1} \parallel$.

3) As introduced above, a particle choosing its learning weights and learning objective dimensions from different exemplars is based on its fitness. However, it is not to say that the particle need to tune its learning model in each generation since we regard that adjusting the learning wights and objective dimensions too frequently may disturb the population's current promising search direction. On the contrary, if a particle cannot reselect its learning model timely, the population may waste many evaluations on its current fitness landscape. In this research, we select the stagnation of $\mathbf{Gb}$ as a criterion for the population to

readjust its learning model. The motivation of the strategy is that individuals in the population can not find out more promising regions if they still adopt the current learning mechanism. Thus, in this work, if **Gb** has stagnated, all individuals in the population will adjust their learning weights and objective dimensions vectors according to their roles. We call the adjustment as "global tuning". In addition, it is another situation need to be considered that some individuals has stagnated though **Gb** is still improved. In this case, all individuals reselecting their learning model may disturb the population's current favorable searching direction. To give those stagnated particles new search mechanisms without hampering the population's searching direction, we only choose those stagnated particles to carry out the adjustment operator. Specifically, only those stagnated individuals will tune their objective-dimension vectors based on their roles. We call the adjustment as "local tuning". According to the discussions above, the adjustments of learning weights and objective dimensions vectors are defined as (4) and (5), respectively.

$$[c_1^i, c_2^i, c_3^i] = \begin{cases} [c_h, c_m, c_l] & \text{if the } i\text{th particle is a leader} \\ [c_l, c_m, c_h] & \text{if the } i\text{th particle is a rambler} \\ [c_l, c_h, c_m] & \text{if the } i\text{th particle is a follower} \end{cases} \tag{4}$$

$$[s_{d1}^i, s_{d1}^i, s_{d1}^i] = \begin{cases} [s_{dh}, s_{dm}, s_{dl}] & \text{if the } i\text{th particle is a leader} \\ [s_{dl}, s_{dm}, s_{dh}] & \text{if the } i\text{th particle is a rambler} \\ [s_{dl}, s_{dh}, s_{dm}] & \text{if the } i\text{th particle is a follower} \end{cases} \tag{5}$$

4) Considering that the exploitative ability is beneficial for improving solutions' accuracy, we adopt BFGS Quasi-Newton method, which has been applied in many PSO variants [18, 42], as a local-searching operator at the later stage of evolution. During the later stage of evolutionary, only **GBest** is selected to carry out the local-searching process. In the local-searching operator, we assign $[0.1 * MaxFEs]$ evaluations to refine **GBest** based on BFGS Quasi-Newton method. To simplify, the function "$fminunc$" in Matlab 2009a is employed in

this research to realize the BFGS Quasi-Newton local-searching operator.

Together with the aforementioned characteristics, the pseudocode and flowchart of FMPSO are detailed in Algorithm 1 and Fig. 1, respectively.

Algorithm 1. FMPSO ()

/* $Stag_i$ and $Stag_g$ represent the stagnated generations of the ith particle and the entire population; */
/* p_e, p_o, and p_{so} are ratios of leader, rambler, and follower, respectively; */
/* c_1^i, c_2^i and c_3^i indicate the ith particle's learning weights for the three exemplars, respectively; */
/* s_{d1}^i, s_{d2}^i, and s_{d3}^i represent the ith particle's for the three exemplars, respectively; */
Begin:
Initialize $\mathbf{X}_i$ and $\mathbf{V}_i$ for N individuals;
Initialize $MaxFEs$; Set $fes = 0$, $Stag_i = 0$, $Stag_g = 0$;
Initialize c_h, c_m, c_l, s_{dh}, s_{dm}, s_{dl}, p_e, p_o, and p_{so};
Evaluate $\mathbf{X}_i$ $(1 \le i \le N)$; $fes = fes + N$;
Update $\mathbf{Pb}_i$ and $\mathbf{Gb}$ according to the finesses.
Initialize c_1^i, c_2^i, c_3^i, s_{d1}^i, s_{d2}^i and s_{d3}^i for each particle according to (4) and (5);

While $(fes < 0.9 * MaxFEs)$

  For $(i = 1$ to $N)$ /* particles' fly */
    Update $\mathbf{V}_i$ and $\mathbf{X}_i$ according to (3) and (2), respectively;
    Evaluate $\mathbf{X}_i$; $fes = fes + 1$;
    If $fit(\mathbf{X}_i) < fit(\mathbf{Pb}_i)$ /* Update $\mathbf{Pb}_i$ */
      $\mathbf{Pb}_i = \mathbf{X}_i$; $fit(\mathbf{Pb}_i) = fit(\mathbf{X}_i)$; $Stag_i = 0$;
    Else
      $Stag_i = Stag_i + 1$;
    End If
  End For

  TmpGb $= \{x | fit(x) = \min(fit(\mathbf{Pb}_i)), 1 \le i \le N\}$; /* Update **Gb** */
  If $fit($**TmpGb**$) < fit($**Gb**$)$
    Gb = **TmpGb**; $fit($**Gb**$) = fit($**TmpGb**$)$; $Stag_g = 0$;
  Else
    $Stag_g = Stag_g + 1$;
  End If

  If $(Stag_g > 0)$ /* global tuning */
    Sort all individuals meeting the condition $fit(\mathbf{x}_{j_1}) \le fit(\mathbf{x}_{j_2}) \le \ldots \le fit(\mathbf{x}_{j_N})$;
    Assign three roles, i.e., leader, rambler, and follower for all individuals according to p_e, p_o, and p_{so};
    Update each individual's learning weights and objective dimensions vectors according to (4) and (5), respectively;
    $Stag_g = 0$;
  Else /* local tuning */
    If (Any individual is stagnated)
      Sort all individuals meeting the condition $fit(\mathbf{x}_{j_1}) \le fit(\mathbf{x}_{j_2}) \le \ldots \le fit(\mathbf{x}_{j_N})$;
      Assign three roles, i.e., leader, rambler, and follower for all individuals according to p_e, p_o, and p_{so};
      For $(i = 1$ to $N)$
        If $(Stag_i > 0)$
          Update the ith particle's objective dimensions vectors according to (5);
          $Stag_i = 0$;
        End If
      End For
    End If
  End If

End While
Assign $0.1 * MaxFEs$ for **Gb** to carry out the local-searching operator.
End.

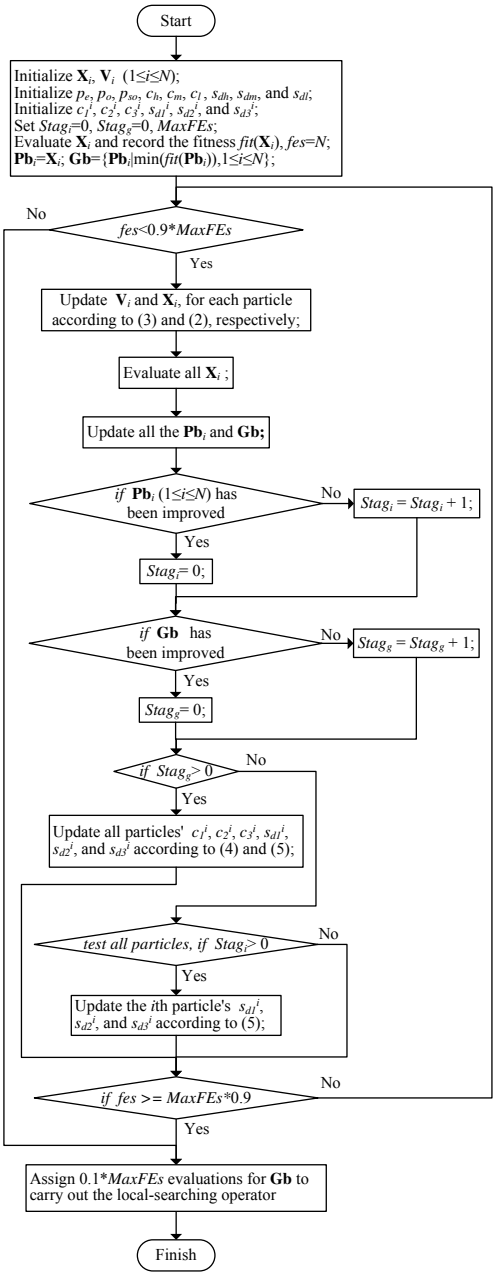


Figure 1: Flowchart of FMPSO algorithm.

In FMPSO, there are some new introduced parameters (see the line03 in Algorithm 1). From a few trail-and-error experiments, we select a set of values for these parameters, the details of which are listed in Table 1.

4. Experimental results and discussions

4.1. Peer algorithms

To verify the performance of FMPSO, some PSO variants based on time-varying parameters and/or updating learning mechanism are selected in this research. All the selected peer algorithm are proposed in the last decade, including CLPSO [31], Frankenstein's PSO [15], OLPSO [44], SLPSO [16], QPSO[32], PSODDS [45], SL_PSO [34], HCLPSO [33], and EPSO [46].

In the first peer algorithm CLPSO, different dimensions of a particle have different exemplars. Furthermore, different learning probabilities assigned for different particles cause the population has different levels of exploration and exploitation ability. These characteristics enable the population in CLPSO to deal with diverse problems. Many experiments verify the better performance of CLPSO on multimodal functions. In Frankenstein's PSO, different components, including time-varying population topology, velocity-update mechanism, and time-decreasing inertia weight, in various PSO variants are integrated into a single framework. The motivation behind the algorithm is to take advantages various merits of different components. In many cases, Frankenstein's PSO is capable of yielding more favorable results than the variants from which its components were taken. The third is OLPSO in which an orthogonal learning (OL) strategy is introduced to discover more useful knowledge lied in individuals' experience. Some experiments indicate that the OL strategy can direct particles to fly towards promising regions by constructing an efficient exemplar. SLPSO is the fourth peer algorithm in which each particle has a set of four candidate operators. In each generation, a particle in SLPSO can select a proper operator to cope with different situations according to the particle's experience. Experiments on some challenging problems manifest the efficiency and effectiveness of

the self-learning mechanism. In QPSO, particles of PSO with quantum behavior to move in a potential field that can ensure the population with promising explorative capability. In the sixth algorithm, i.e., PSODDS, the random coefficients are removed from particles' velocity update rule. Moreover, each particle only chooses those dimensions that are far away from the **Gb**'s corresponding dimensions as its learning objective dimensions. The dimensions selection method can help the population to keep the diversity, and then to prevent the premature convergence. In SL_PSO, a novel social learning mechanism is introduced into PSO. Based on the social learning mechanism, each individual in SL_PSO learns from any better particles termed as demonstrators in the current swarm. Another advantage of SL_PSO is a dimension-dependent parameter control method which cause it relieve the burden of parameter settings. Some comparative results demonstrate that it performs well on low-dimensional problems and is favorable for solving large-scale functions as well. HCLPSO is a CLPSO variant in which the entire population is divided into two subpopulations aiming to focus on explorative and exploitation, respectively. The extensive experiments verify that the heterogeneous construction can offset the contradiction between the global search ability and local search capability. EPSO is the last peer algorithm, in which five PSO variants including inertia weight PSO, CLPSO, FDR-PSO, HPSO-TVAC, and LIPS are hybridized to complement each other. To take full advantages of all algorithms, a self-adaptive mechanism is introduced in EPSO.

The basic configurations of all the peer algorithms listed in Table 1 are exactly the same as that used in the original literatures.

4.2. Experiment 1: Comparison results on CEC2005 test suite

In the first experiment, a popular test suite, i.e., CEC2005 test suite, is used to evaluate the performance of the peer algorithms. In the test suite, the 25 functions (f_1-f_{25}) can be divided into four groups according to the basic characteristics, including unimodal functions (f_1-f_5), basic multimodal functions (f_6-f_{12}), expanded multimodal functions (f_{13}-f_{14}), and hybrid composition multimodal

Table 1: Parameters settings of the 10 algorithms

Algorithm	Year	Parameters Settings
CLPSO	2006	$w = 0.9 - 0.5 * gen/MaxGen$,$c$=1.49445
		$Pc_i = 0.05 + 0.45 * (exp(10 * (i-1)ps - 1) - 1)/(exp(10) - 1)$
*F_PSO	2009	w=:[0.4,0.9], $Tmax = 3 * N^2$, $K = 4 * N$
OLPSO	2011	w:[0.4,0.9], c=2.0, G=5
SLPSO	2012	w:[0.4,0.9], η=1.496, γ=0.01
QPSO	2012	$\alpha = 0.96$
PSODDS	2013	χ=0.7298, c_1=c_2=2.05
SL_PSO	2015	$N = 100 + \lfloor D/10 \rfloor$, $\alpha = 0.5$, $\beta = 0.01$
HCLPSO	2015	w=0.99∼0.2, c_1=2.5∼0.5, c_2=0.5∼2.5, c=3∼1.5
#EPSO	2017	-
FMPSO	–	$c_h = 1.2$, $c_m = 0.7$, $c_l = 0.2$,
		$\parallel s_{dh} \parallel = 0.6 * D$, $\parallel s_{dm} \parallel = 0.3 * D$, $\parallel s_{dl} \parallel = 0.1 * D$

* To describe simply, Frankenstein's PSO is renamed as F_PSO in this study.

There are too many parameters involved in EPSO. Due to the space limitation, more detailed information of the parameters setting one can refer to [46].

functions (f_{15}-f_{25}).

The number of decision variables is set to D=30 for all the benchmark functions. Due to the space limitation, the details of CEC2005 test suite can refer to the literature [48].

To fairly compare the performance among the 10 algorithms, each peer algorithm carried out 30 independent runs on each test function. The maximum number of fitness evaluations ($MaxFEs$) for each run is set as 300 000. The size of population is $N = 60$ for FMPSO. Other peer algorithms' population sizes are the same as the original literatures.

4.2.1. Comparison on solution accuracy

In this section, experiments are conducted to compare the performance of each algorithm in terms of the mean and standard deviations of the solutions. The corresponding results are provided in Table 2, where the best result on each function among all algorithms is shown in bold. In the last two rows, "(#)♠+" and "(#)♠-" represent the number that FMPSO is significant better than and significant worsen than the corresponding peer algorithm, respectively. "(#)Best-Mean" denotes the number of the best mean values achieved by FMPSO.

Table 2: Comparison results of solution accuracy on CEC2005 test suite.

		CLPSO	FPSO	OLPSO	SLPSO	QPSO	PSODDS	SL_PSO	HCLPSO	EPSO	FMPSO
	Mean	**0.00E+00**	3.65E-27	**0.00E+00**	**0.00E+00**	9.43E-27	2.22E+02	**0.00E+00**	**0.00E+00**	2.37E-26	**0.00E+00**
f_1	Std.Dev	0.00E+00	3.46E-28	0.00E+00	0.00E+00	1.33E-27	2.15E+02	0.00E+00	0.00E+00	9.21E-27	0.00E+00
	p-value	=	♠0.000+	=	=	♠0.000+	♠0.000+	=	=	♠0.000+	
	Mean	1.97E+02	9.53E-07	3.93E+04	7.16E+00	3.53E-02	1.34E+02	3.75E-11	6.91E-05	2.29E-13	**4.37E-25**
f_2	Std.Dev	2.90E+01	5.37E-07	9.81E+03	5.09E+00	2.88E-02	2.05E+02	4.67E-11	4.07E-05	1.71E-13	4.65E-25
	p-value	♠0.000+	♠0.000+	♠0.000+	♠0.000+	♠0.000+	0.078+	♠0.011+	♠0.000+	♠0.000+	
	Mean	1.39E+07	2.79E+06	3.18E+07	2.49E+06	2.72E+06	1.53E+06	6.13E+05	1.02E+06	**2.74E+05**	1.10E+06
f_3	Std.Dev	2.08E+06	9.65E+05	9.48E+06	1.22E+06	7.91E+05	1.62E+06	2.18E+05	6.57E+05	8.72E+04	7.34E+05
	p-value	♠0.000+	♠0.000+	♠0.000+	♠0.000+	0.734+	0.352+	♠0.019-	0.051-	♠0.000-	
	Mean	3.08E+03	2.15E+02	3.76E+04	1.31E+04	1.53E+02	1.52E+04	**5.96E-02**	7.82E+02	3.26E+02	1.43E+03
f_4	Std.Dev	5.38E+02	1.12E+02	5.72E+03	2.12E+03	8.80E+01	4.93E+03	7.03E-02	2.95E+02	2.24E+02	6.75E+02
	p-value	♠0.000+	♠0.000-	♠0.000+	♠0.000+	♠0.000-	♠0.000+	♠0.000-	♠0.000-	♠0.000-	
	Mean	3.22E+03	2.87E+03	2.85E+03	6.38E+03	2.47E+03	5.38E+03	**2.30E+03**	3.17E+03	4.66E+03	4.17E+03
f_5	Std.Dev	2.18E+02	4.65E+02	1.57E+03	7.98E+02	5.12E+02	1.22E+03	3.59E+02	4.89E+02	4.14E+02	1.25E+03
	p-value	♠0.002-	♠0.000-	♠0.005-	♠0.000+	♠0.000-	♠0.004+	♠0.000-	♠0.000-	♠0.000+	
	Mean	4.91E+01	4.78E+01	1.12E+02	5.80E+01	5.83E+01	5.82E+05	9.97E+01	3.19E+00	3.81E+00	**9.44E-01**
f_6	Std.Dev	2.53E+01	3.55E+01	3.42E+01	4.03E+01	3.92E+01	1.05E+06	1.25E+02	3.03E+00	3.45E+00	1.42E+00
	p-value	♠0.000+	♠0.000+	♠0.000+	♠0.000+	♠0.000+	0.266+	♠0.017+	♠0.019+	♠0.000+	
	Mean	9.91E-02	1.85E-02	2.28E+00	2.31E-02	2.71E+03	1.34E+01	2.15E-02	1.59E-02	1.96E-02	**1.29E-02**
f_7	Std.Dev	4.12E-02	1.19E-02	1.22E+00	1.42E-02	0.00E+00	6.82E+00	1.13E-02	1.06E-02	1.32E-02	9.41E-03
	p-value	♠0.000+	0.117+	♠0.000+	♠0.026+	♠0.000+	♠0.000+	♠0.011+	0.135+	♠0.000+	
	Mean	2.09E+01	2.09E+01	2.11E+01	2.00E+01	2.10E+01	2.00E+01	**1.89E+01**	2.03E+01	2.09E+01	1.97E+01
f_8	Std.Dev	6.95E-02	2.86E-02	4.17E-02	2.32E-02	3.28E-02	5.11E-02	3.77E+00	1.41E-01	2.30E-02	1.31E+00
	p-value	0.094+	0.082+	♠0.049+	0.669+	0.051+	0.651+	0.532-	0.071+	♠0.049+	
	Mean	**0.00E+00**	3.41E+01	9.89E-01	6.40E-14	1.55E+01	7.08E+01	3.27E+01	3.32E-02	1.18E-16	3.50E+01
f_9	Std.Dev	0.00E+00	6.94E+00	1.38E+00	1.18E-13	2.47E+00	1.32E+01	1.17E+00	6.41E-02	2.21E-16	7.50E+00
	p-value	♠0.000-	0.698-	♠0.000-	♠0.000-	♠0.000-	♠0.000+	0.445-	♠0.000-	♠0.000-	
	Mean	9.33E+01	1.72E+02	1.65E+02	9.40E+01	6.49E+01	1.51E+02	1.60E+02	5.38E+01	5.51E+01	**4.75E+01**
f_{10}	Std.Dev	1.16E+01	9.87E+00	4.29E+01	2.00E+01	3.69E+01	2.90E+01	7.27E+00	9.80E+00	1.48E+01	5.01E+00
	p-value	♠0.000+	♠0.000+	♠0.000+	♠0.000+	♠0.000+	♠0.000+	♠0.000+	♠0.044+	♠0.038+	
	Mean	2.67E+01	1.15E+01	2.41E+01	2.86E+01	3.97E+01	2.32E+01	**6.22E+00**	2.11E+01	2.30E+01	1.90E+01
f_{11}	Std.Dev	1.01E+00	2.41E+00	2.18E+00	2.11E+00	6.84E-01	3.03E+00	1.45E+00	2.76E+00	1.79E+00	2.69E+00
	p-value	♠0.000+	♠0.000-	♠0.000+	♠0.000+	♠0.000+	♠0.000+	♠0.000-	0.596+	0.128+	
	Mean	1.96E+04	8.91E+03	3.63E+04	**4.44E+03**	9.15E+05	2.53E+04	6.50E+03	4.49E+03	5.05E+03	2.12E+04
f_{12}	Std.Dev	3.26E+03	7.36E+03	1.17E+04	3.36E+03	1.13E+05	1.90E+04	5.45E+03	2.93E+03	3.86E+03	1.65E+04
	p-value	0.628-	♠0.005-	♠0.003+	♠0.000-	♠0.000+	0.487+	♠0.001-	♠0.000-	♠0.000-	
	Mean	3.81E+00	1.10E+01	2.41E+00	**1.04E+00**	2.66E+00	3.77E+00	3.54E+00	2.32E+00	1.89E+00	2.74E+00
f_{13}	Std.Dev	2.31E-01	1.15E+00	3.19E-01	1.54E-01	3.91E-01	7.79E-01	4.73E-01	1.90E-01	3.18E-01	8.13E-01
	p-value	♠0.000+	♠0.000+	0.074-	♠0.000-	♠0.000-	♠0.000+	♠0.000+	♠0.000-	♠0.000-	
	Mean	1.29E+01	1.22E+01	1.37E+01	1.29E+01	1.26E+01	1.29E+01	1.30E+01	1.21E+01	1.23E+01	**1.19E+01**
f_{14}	Std.Dev	1.49E-01	1.99E-01	1.67E-01	2.35E-01	3.26E-01	3.21E-01	1.80E-01	3.77E-01	7.01E-01	4.24E-01
	p-value	♠0.000+	♠0.022+	♠0.000+	♠0.000+	0.051+	♠0.000+	♠0.000+	♠0.009+	0.581+	
	Mean	1.33E+02	3.19E+02	2.66E+02	2.94E+02	7.61E+02	4.89E+02	4.04E+02	9.21E+01	**8.29E+01**	3.18E+02
f_{15}	Std.Dev	5.99E+01	4.37E+01	5.50E+01	1.03E+02	4.68E+01	3.76E+01	7.68E+01	8.73E+01	9.08E+01	9.02E+01
	p-value	♠0.000-	0.967+	♠0.034-	0.478-	♠0.002+	♠0.000+	0.845+	0.780-	0.987-	
	Mean	1.18E+02	3.78E+02	1.62E+02	1.97E+02	9.03E+02	3.57E+02	1.43E+02	1.38E+02	1.55E+02	**1.04E+02**
f_{16}	Std.Dev	1.97E+01	1.58E+02	1.87E+01	8.40E+01	1.24E+02	1.23E+02	1.33E+02	4.53E+01	4.75E+01	2.30E+01
	p-value	0.064+	♠0.000+	♠0.000+	♠0.000+	♠0.000+	♠0.000+	♠0.022+	♠0.000+	♠0.000+	
	Mean	1.73E+02	4.08E+02	2.02E+02	2.71E+02	1.22E+03	3.63E+02	2.84E+02	2.51E+02	1.71E+02	**1.65E+02**
f_{17}	Std.Dev	2.84E+01	1.15E+02	2.07E+01	1.31E+02	1.07E+02	1.63E+02	2.34E+02	7.55E+01	1.65E+01	6.52E+01
	p-value	0.694+	♠0.000+	0.070+	♠0.002+	♠0.042+	♠0.000+	♠0.003+	♠0.044+	♠0.000+	
	Mean	8.06E+02	9.00E+02	8.13E+02	8.34E+02	9.37E+02	1.08E+03	8.44E+02	8.50E+02	1.42E+03	**8.00E+02**
f_{18}	Std.Dev	4.82E+01	0.00E+00	3.48E+01	1.51E+02	1.63E+02	1.75E+02	7.57E+01	1.23E+02	4.45E+01	0.00E+00
	p-value	0.776+	♠0.000-	0.232+	0.415+	♠0.000+	♠0.000+	0.057+	♠0.000+	♠0.000+	
	Mean	**7.33E+02**	9.00E+02	7.97E+02	8.10E+02	9.86E+02	1.01E+03	8.14E+02	8.73E+02	9.57E+02	8.00E+02
f_{19}	Std.Dev	1.78E+02	0.00E+00	9.10E+00	2.45E+02	1.98E+02	1.80E+02	2.70E+01	2.42E+02	9.18E+01	0.00E+00
	p-value	0.116-	♠0.000+	0.614-	0.855+	♠0.000+	♠0.000+	0.321+	♠0.000+	♠0.000+	
	Mean	7.40E+02	9.00E+02	7.78E+02	8.69E+02	1.05E+03	1.19E+03	8.21E+02	9.28E+02	9.22E+02	**7.00E+02**
f_{20}	Std.Dev	1.31E+02	0.00E+00	4.91E+01	1.12E+02	1.80E+02	1.04E+02	1.03E+02	1.07E+02	3.89E+00	1.33E+02
	p-value	0.323+	♠0.000+	♠0.000+	♠0.000+	♠0.001+	♠0.000+	♠0.009+	♠0.041+	♠0.000+	
	Mean	5.00E+02	**4.88E+02**	1.05E+03	6.30E+02	8.66E+02	8.38E+02	5.00E+02	5.00E+02	6.96E+02	9.26E+02
f_{21}	Std.Dev	0.00E+00	2.10E+01	4.31E+01	1.99E+02	2.96E+00	2.90E+02	0.00E+00	2.80E-13	2.75E+02	2.44E+02
	p-value	♠0.000-	♠0.000-	♠0.012+	♠0.000-	♠0.000-	0.243-	♠0.000-	♠0.000-	0.372-	
	Mean	9.49E+02	9.26E+02	9.00E+02	9.83E+02	**5.58E+02**	9.88E+02	8.93E+02	9.11E+02	9.58E+02	8.87E+02
f_{22}	Std.Dev	6.58E+00	1.47E+01	1.47E+01	2.50E+01	5.31E+01	5.74E+01	1.05E+01	1.22E+01	1.49E+01	2.70E+01
	p-value	♠0.000+	♠0.000+	0.073+	♠0.000+	0.067-	♠0.000+	0.407+	♠0.002+	♠0.013+	
	Mean	**5.34E+02**	6.18E+02	1.07E+03	6.11E+02	8.71E+02	1.02E+03	5.38E+02	**5.34E+02**	7.01E+02	8.79E+02
f_{23}	Std.Dev	4.11E-05	1.37E+02	2.55E+01	1.33E+02	2.32E+00	1.58E+02	7.07E+00	3.56E-04	2.34E+02	2.76E+02
	p-value	♠0.000-	♠0.000-	♠0.001+	♠0.000-	♠0.000-	♠0.029+	♠0.000-	♠0.000-	0.095-	
	Mean	**2.00E+02**	2.08E+02	9.41E+02	3.17E+02	2.20E+02	1.04E+03	**2.00E+02**	2.98E+02	3.24E+02	5.48E+02
f_{24}	Std.Dev	9.76E-14	1.32E+01	3.08E+00	2.03E+02	2.04E+00	1.36E+02	0.00E+00	1.39E+00	2.07E+02	3.58E+02
	p-value	♠0.000-	♠0.000-	♠0.000+	♠0.000-	♠0.000-	♠0.000+	♠0.000-	♠0.000-	♠0.000-	
	Mean	**2.00E+02**	**2.00E+02**	9.41E+02	4.16E+02	2.20E+02	1.06E+03	**2.00E+02**	2.98E+02	3.49E+02	5.75E+02
f_{25}	Std.Dev	0.00E+00	1.17E-17	3.41E+00	3.31E+02	1.71E+00	1.10E+02	0.00E+00	1.33E+00	2.39E+02	3.75E+02
	p-value	♠0.000-	♠0.000-	♠0.000+	0.121-	♠0.000-	♠0.000+	♠0.000-	♠0.000-	♠0.000-	
(#)	Best-Mean	6	2	1	3	1	0	7	2	2	10
(#)	♠+	10	12	16	13	13	19	9	10	13	
(#)	♠ -	7	9	3	6	8	0	9	9	7	

To describe simply, Frankenstein's PSO is renamed as FPSO.

♠: The difference between the two samples (p-value) is significant at level α=0.05 by the t-test.

+, -, =: FMPSO is better than, worse than, or the same as the corresponding compared algorithm, respectively.

(#): The number of the corresponding index.

1) *Unimodal Functions* (f_1-f_5): For the five unimodal functions, the results indicate that SL_PSO yields the most favorable performance since it achieves the best solutions on 3 out of the 5 problems, followed by FMPSO and EPSO who obtain the best results on 2 and 1 functions, respectively. For the very simple unimodal problem f_1, there are 6 algorithms offer the global optimal solution. However, for those non-separable unimodal problems, all the peer algorithms have not found out the global optimal solutions since some complex correlations between variables can not be captured by these algorithms. For the problem f_2, FMPSO yields the best solution which is significantly better than the results obtained by other peer algorithms. However, when the Gaussian noise is added into f_2, (i.e., f_4) it does not achieve the best result on the modified function. On the contrary, SL_PSO shows very promising performance on f_4. The comparison results demonstrate that the capability of FMPSO for dealing with noise functions need to be further improved.

2) *Multimodal Functions* (f_6-f_{14}): In FMPSO, each particle adjusts its role based on its fitness, the aim of which is to effectively balance the contradiction between explorative and exploitative capability which is a crucial issue in multimodal problems optimization. Indeed, the experimental results on the 9 multimodal functions given in Table 2 (i.e., f_6-f_{14}) do support this intuition. FMPSO surpasses all the other peer algorithms in terms of the number of the achieved best results. Specifically, FMPSO yields the best result on 4 out of the 9 multimodal problems, followed by SL_PSO who yields the best results on 2 problems. On the separable multimodal problem f_9, only CLPSO and SLPSO demonstrate very reliable performance since they yield the global optimal solutions on some runs. On the other non-separable multimodal problems, all the peer algorithms have not achieved the global optimum solutions. Although FMPSO, SL_PSO, and SLPSO offer the best results on these non-separable functions, it seems that there is no significant different between the results since the difference among the results is very small except FMPSO yields a significant better result than other algorithms on f_6. The favorable performance of FMPSO on f_6 where there is a very narrow valley from local optimum to global optimum

indicates that the multi-role mechanism in it may enable the population with a very reliable performance on global searching ability.

3) *Hybrid Composition Functions* (f_{15}-f_{25}): for the 11 hybrid composition functions, FMPSO offers the best performance on f_{16}, f_{17}, f_{18}, and f_{20}, in terms on convergence speed as well as solution accuracy. Moreover, it also shows very promising performance on f_{19}. CLPSO and SL_PSO, as other two excellent algorithms, achieve almost the same best property on f_{21}, f_{23}, f_{24}, and f_{25} except the former algorithm has slower convergence speed than the latter one at the initial evolutionary stage. In addition, CLPSO also yields a favorable performance on f_{15} since it has a stable convergence process during the entire optimization process. It is worth to note that FMPSO obtains very adverse results on f_{21}, f_{23}, f_{24}, and f_{25} since its population has been trapped into local optimum during the early evolutionary stage. The comparison results on the hybrid composition functions indicate that how to draw much useful information of an objective problem's characteristics is a crucial issue for optimization of it.

4.2.2. Statistical test results

Generally, it is necessary to use statistical tests to analyze the experimental results. In this section, two tests, i.e., two-tailed t-test and Friedman-test, are carried out to validate the performance of all the algorithms.

The t-test is a popular parametric statistical hypothesis test when test statistics follow a normal distribution. For example, it can be used to determine whether two sets of data are significantly different from each other. In this work, the t-test is used to testify whether FMPSO is significantly better or not than a peer algorithm on a test function. In other words, it is applied to compare the performance between FMPSO and a peer algorithm on a specific function.

The Friedman-test is a non-parametric statistical test. Generally, it is used to detect differences in treatments across multiple test attempts. The procedure involves ranking each row (or block) together, then considering the values of ranks by columns. In our study, the Friedman-test is adopted to compare the comprehensive performance of each algorithm on a set of functions. Thus, the

result of the Friedman-test provides an overview of algorithms' performance.

Note that the results of t-test are also presented in Table 2 while the results of Friedman-test are listed in Table 3, in which each algorithm and its rankings are listed in ascending order (the lower the better). Furthermore, we also separately carry out Friedman-test of all peer algorithms on unimodal, multimodal, and hybrid composition functions, the results of which also listed in Table 3. The statistic and the corresponding p values are shown at the bottom of Table 3.

1) t-test results: If the p-value obtained in the hypothesis test is less than the significance level α=0.05, the difference of results is statistically significant. It can be seen from Table 2 that FMPSO and SL_PSO have the same performance because the numbers of the problems they dominate each other are both 9 although FMPSO yields the best mean results on 10 functions that is slightly better than that of SL_PSO. Moreover, CLPSO also attains competitive results in this perspective. In the unimodal problems, SL_PSO surpasses FMPSO on 3 functions while it is dominated by FMPSO only on f_2. However, FMPSO demonstrates the best results on the 4 multimodal functions while the maximum number of the best results that other algorithms dominated it is only 2. Furthermore, FMPSO significantly dominates other three outstanding algorithms, i.e., SL_PSO, CLPSO, and HCLPSO, on 5, 6, and 3 multimodal problems respectively, in terms of the hypothesis test results. The results may verify the efficiency of the multi-role introduced in FMPSO for dealing with difficult fitness landscape. However, CLPSO and SL_PSO display more favorable performance on the hybrid composition functions than FMPSO.

2) Friedman-test results: From the test results listed in Table 3, we can see that FMPSO attains the best comprehensive performance on the CEC2005 test suite, followed by HCLPSO, SL_PSO, CLPSO, and SLPSO. In addition, SL_PSO, HCLPSO, and CLPSO achieve the most favorable results on the unimodal, multimodal, and hybrid composition functions, respectively. Although FMPSO does not yield the best results on the different categories problems, it displays very reliable and stable performance on the different problems since FMPSO shows the second best performance on the three types problems.

Table 3: Friedman-test on CEC2005 test suite

Average Rank	Algorithm	Ranking	Unimodal: f_1-f_5		Multimodal: f_6-f_{14}		Hybird: f_{15}-f_{25}	
			Algorithm	Ranking	Algorithm	Ranking	Algorithm	Ranking
1	FMPSO	3.86	SL_PSO	2.10	HCLPSO	2.89	CLPSO	2.55
2	HCLPSO	3.88	FMPSO	4.30	FMPSO	3.44	FMPSO	4.00
3	SL_PSO	4.32	HCLPSO	4.30	EPSO	3.67	SL_PSO	4.32
4	CLPSO	4.70	EPSO	4.80	SLPSO	4.72	HCLPSO	4.50
5	EPSO	4.92	QPSO	5.00	SL_PSO	5.56	SLPSO	5.73
6	SLPSO	5.60	FPSO	5.20	FPSO	5.78	OLPSO	5.73
7	FPSO	5.64	SLPSO	6.90	CLPSO	6.11	FPSO	5.73
8	OLPSO	6.82	CLPSO	6.90	QPSO	7.33	EPSO	6.00
9	QSPO	6.88	OLPSO	7.30	PSODDS	7.61	QPSO	7.36
10	PSODDS	8.38	PSODDS	6.20	OLPSO	7.89	PSODDS	9.09
Statistic	54.312		16.929		28.634		36.379	
p value	0.000		0.050		0.001		0.000	

To describe simply, Frankenstein's PSO is renamed as FPSO.

From the comparison results introduced above, a tentative conclusion can be observed that the random selected learning exemplars and objective dimensions for different particles may not help for FMPSO to capture the essential characteristics implied in the composition problems. Furthermore, when to execute the "global tuning" and "local tuning" may be another issue to be deal with if we want to achieve a nice tradeoff between the explorative and exploitative capabilities. Due to the space limitation, these issues will be study on our future works.

4.2.3. Comparison on convergence speed

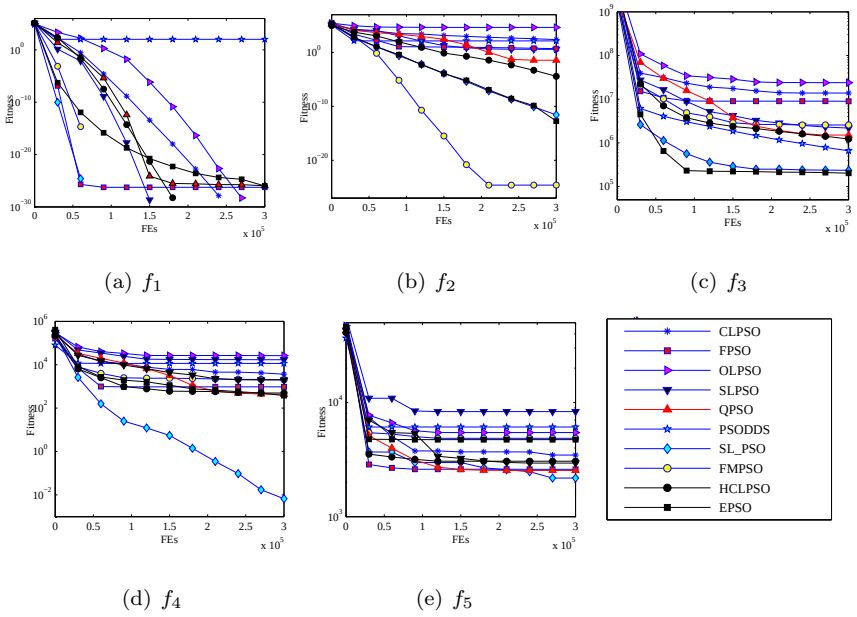
The prime motivation of the local-searching strategy is to improve the accuracy of solutions. In this section, experiments are conducted to compare FMPSO with other 6 peer algorithms on convergence speed. The comparison results of convergence process on unimodal functions, multimodal functions, and hybrid composition functions are demonstrated in Fig. 2, Fig. 3, and Fig. 4, respectively.

1) *Unimodal Functions* (f_1-f_5): From the Fig. 2 we can see that FMPSO and SL_PSO have the highest convergence speed on the simple unimodal function (f_1) though some other peer algorithms also achieve the global optimum solution on the problem. On f_2, FMPSO not only obtains the most accurate solution but also displays highest convergence speed, followed by SL_PSO. On the contrary, SL_PSO achieves a favorable performance on f_3 to f_5. Furthermore, EPSO also

obtains a very pleasurable convergence progress on f_3.

2) *Multimodal Functions* (f_6-f_{14}): On the 9 multimodal problems, FMPSO offers the most promising convergence process on f_6, f_7, f_{10}, and f_{14}. Furthermore, FMPSO demonstrates the second best performance on f_8 which is slightly worse than PSODDS, in terms of the convergence speed during the optimization process. Comparing with the results demonstrated in Fig 2, we can see that SL_PSO only displays the highest convergence speed on f_{11} though it achieves the most superior results on the unimodal problems. On the contrary, SLPSO shows very reliable performance on f_7, f_9, f_{10}, and f_{13}, although it can not offer favorable results on the unimodal functions, in terms of the convergence speed. Note that HCLPSO and EPSO display very similar convergence performance on majority of the 9 multimodal functions since the common component of the two algorithm is CLPSO.

3) *Hybrid Composition Functions* (f_{15}-f_{25}): It can be seen from the comparison results on the 11 hybrid composition functions that FMPSO outperforms all the other algorithms on f_{16}, f_{17}, f_{18}, and f_{20}, in terms of the convergence speed as well as the solution accuracy. Furthermore, FMPSO displays almost the same performance as SL_PSO though the former one attains a slightly slower convergence speed at the initial evolutionary stage. Unfortunately, FMPSO offers dreadful performance on f_{21}, f_{23}, f_{24}, and f_{25} since the population of FMPSO has been trapped into local optimum at the early evolutionary stage. On the contrary, CLPSO, HCLPSO and SL_PSO attain pleasurable characteristics, such as accurate solutions and high convergence speed, on those hybrid composition problems. The results verify that the comprehensive learning model has positive effects on complicated problems. Note that QPSO yields very favorable performance on some hybrid composition functions, including f_{18}, f_{19}, f_{22}, f_{24}, and f_{25}, though it does not manifest promising results on the unimodal problems and multimodal problems.



(a) f_1 (b) f_2 (c) f_3

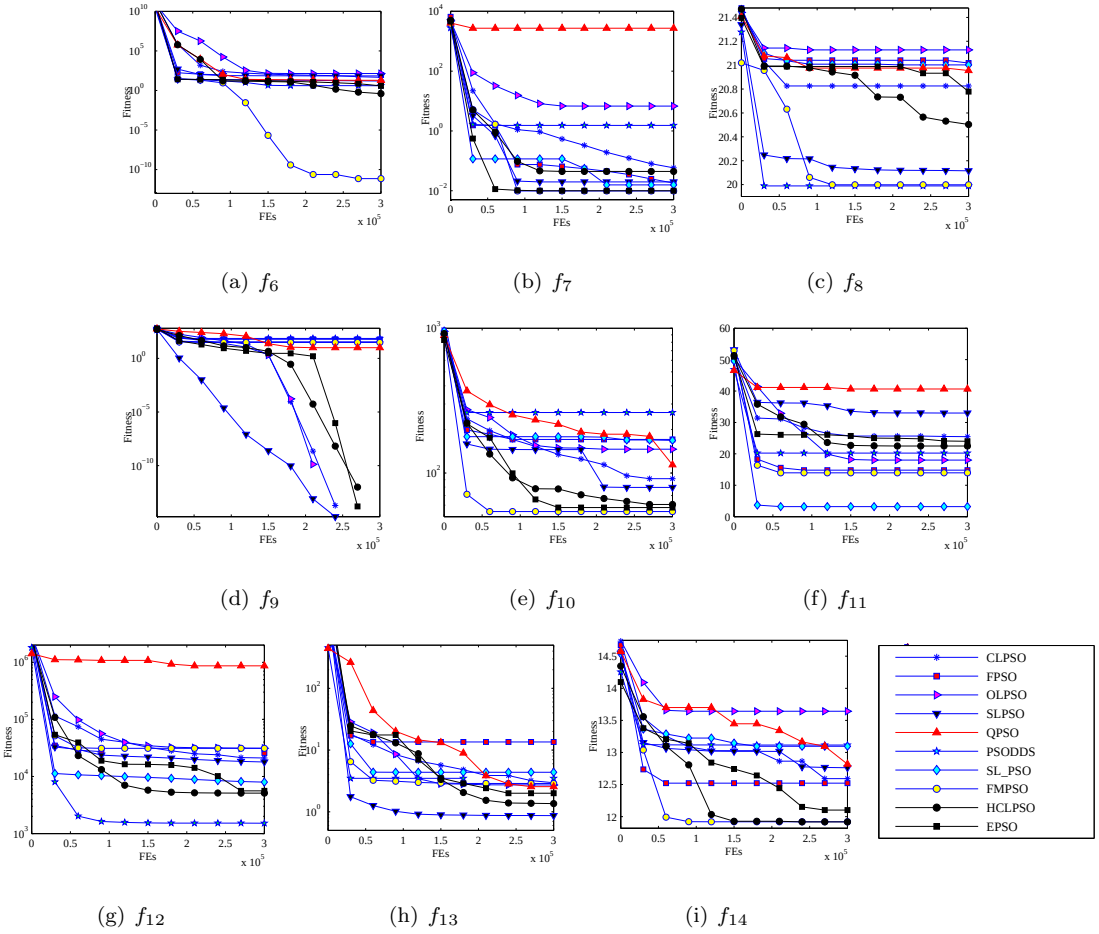
(d) f_4 (e) f_5

Figure 2: Convergence progress of the 10 algorithms on the 5 unimodal functions.

4.2.4. Time usage

In this section, a set of experiments is conducted to compare the time usage of all the peer algorithms since "timing" is of great importance in many real engineering applications. The experimental results of time usage measured by second are listed in Table 4. Note that the metric "*Mean*" in the table denotes the mean time usage of 30 independent runs of an algorithm on a function, and "*Avg*" represents the average time usage of a corresponding algorithm on all test functions.

From Table 4 we can see that Frankenstein's PSO demonstrates the best results on the experiment on all the test problems, in terms of metrics *Mean* and *Avg*. The favorable performance of Frankenstein's is beneficial from that the computational time of some newly added strategies is very little. On the contrary, the time usages of CLPSO, HCLPSO, EPSO and PSODDS are more



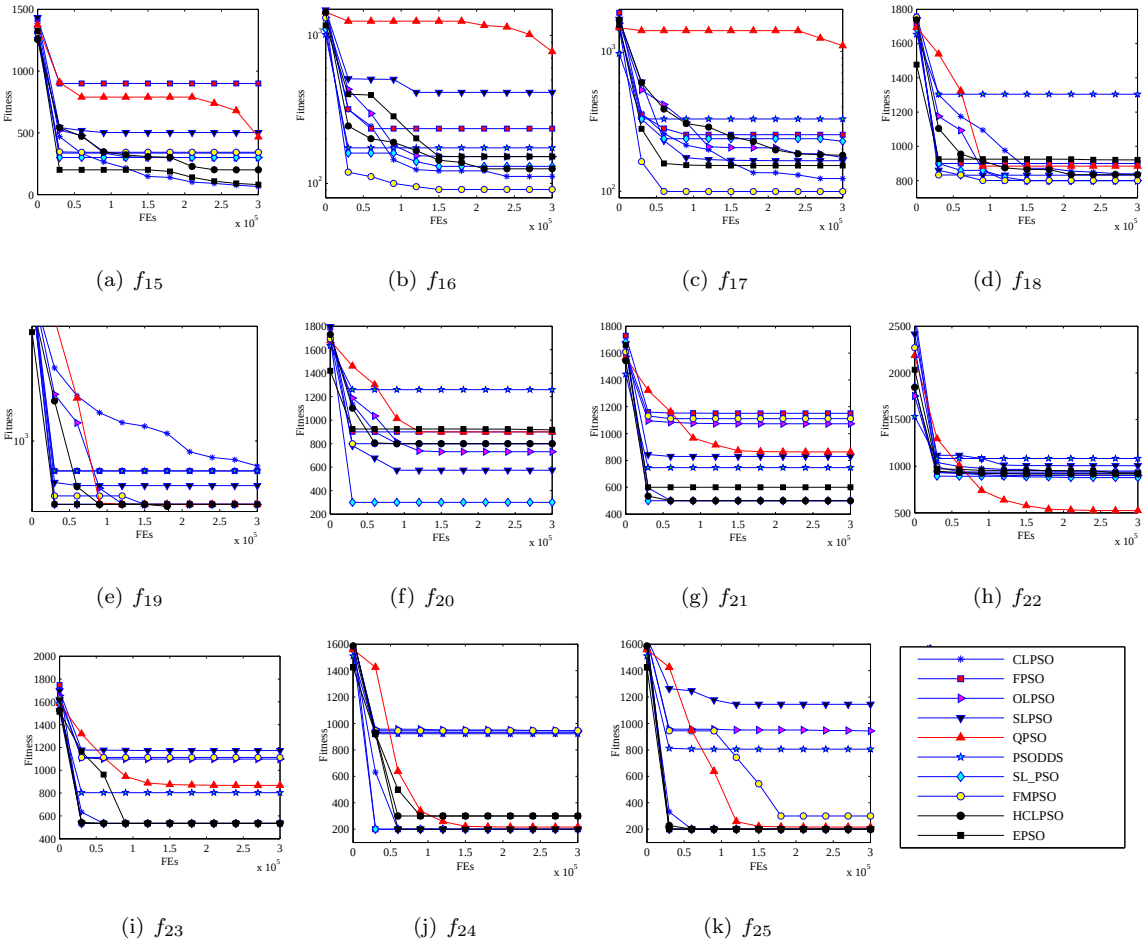
(a) f_6 (b) f_7 (c) f_8

(d) f_9 (e) f_{10} (f) f_{11}

(g) f_{12} (h) f_{13} (i) f_{14}

Figure 3: Convergence progress of the 10 algorithms on the 9 multimodal functions.

than other peer algorithms on the majority of test problems. The comparison results verify that the comprehensive learning model is very time-consuming. Although FMPSO manifests very competitive characteristics on the convergency speed, it achieves an unfavorable performance in terms of time usage since two sort operators (see line 24 and 30 in Algorithm 1) are very time consuming. Although the time complexity of the operators is $O(n\log n)$, they are not carried



(a) f_{15} (b) f_{16} (c) f_{17} (d) f_{18}

(e) f_{19} (f) f_{20} (g) f_{21} (h) f_{22}

(i) f_{23} (j) f_{24} (k) f_{25}

Figure 4: Convergence progress of the 10 algorithms on the 11 hybrid composition functions.

out in each generation. In other words, only if the population and individuals are stagnated, can the population and individuals adjust their learning models. In this condition, FMPSO can archive a proper balance between the solution accuracy and time usage. From the comparison results of Avg, we can see that FMPSO displays the third best performance, which is worse than Frankenstein's PSO and SL_PSO.

Table 4: Time consuming (s) of the 10 peer algorithms on CEC2005 test suite.

		CLPSO	FPSO	OLPSO	SLPSO	QPSO	PSODDS	SL_PSO	HCLPSO	EPSO	FMPSO
f_1	*Mean*	35.46	19.99	25.26	25.72	29.25	33.39	20.80	29.29	49.32	25.14
f_2	*Mean*	50.40	29.11	38.53	36.45	43.76	47.41	34.48	38.21	61.71	37.23
f_3	*Mean*	42.03	22.70	29.20	26.93	33.75	38.04	24.86	31.81	44.88	26.03
f_4	*Mean*	55.99	32.34	43.06	41.65	48.21	51.97	39.04	41.95	58.19	42.20
f_5	*Mean*	33.79	17.56	21.05	19.13	26.98	30.04	17.08	26.56	46.45	20.79
f_6	*Mean*	39.09	21.81	27.64	25.26	34.13	36.31	26.45	30.01	48.54	25.63
f_7	*Mean*	40.86	22.69	29.44	27.60	36.17	37.73	27.00	35.95	49.64	28.04
f_8	*Mean*	43.38	22.65	28.74	26.83	36.67	37.41	27.62	36.20	52.66	28.72
f_9	*Mean*	36.69	20.17	25.26	24.20	32.45	33.76	23.58	38.11	43.61	23.79
f_{10}	*Mean*	38.73	20.94	26.35	24.07	33.78	34.81	24.69	39.76	50.49	25.84
f_{11}	*Mean*	78.26	50.37	70.33	69.41	76.13	80.73	65.62	89.73	111.55	71.53
f_{12}	*Mean*	40.77	24.33	32.07	30.19	36.68	41.88	26.45	37.07	63.88	29.28
f_{13}	*Mean*	47.79	28.66	38.60	37.02	41.06	53.80	33.11	44.43	69.48	38.38
f_{14}	*Mean*	52.23	30.98	41.57	40.84	46.16	59.63	36.93	53.54	86.42	40.77
f_{15}	*Mean*	1551.21	1027.26	1557.92	1577.32	1619.49	1539.81	1555.09	1425.06	1583.37	1545.72
f_{16}	*Mean*	1542.98	1025.41	1601.62	1552.78	1577.97	1528.00	1607.22	1512.75	1587.64	1539.31
f_{17}	*Mean*	1555.76	1030.72	1675.50	1618.46	1561.41	1536.72	1620.72	1516.07	1561.02	1560.68
f_{18}	*Mean*	1541.87	1024.20	1570.89	1628.48	1566.31	1525.34	1533.56	1558.81	1594.50	1541.82
f_{19}	*Mean*	1549.19	1024.78	1589.63	1636.25	1551.83	1526.57	1534.12	1585.25	1630.88	1556.75
f_{20}	*Mean*	1538.88	1021.27	1572.09	1629.44	1541.67	1528.72	1530.05	1596.99	1607.80	1548.53
f_{21}	*Mean*	1608.85	1069.35	1604.83	1692.05	1645.61	1709.19	1587.91	1719.10	1788.52	1618.41
f_{22}	*Mean*	1628.73	1074.66	1608.37	1702.32	1634.32	1735.81	1607.37	1715.91	1861.45	1612.28
f_{23}	*Mean*	1644.23	1075.24	1621.45	1642.51	1634.52	1715.57	1606.52	1686.10	1724.18	1634.61
f_{24}	*Mean*	1552.39	1035.42	1547.75	1606.49	1570.58	1569.36	1532.73	1591.10	1654.46	1570.43
f_{25}	*Mean*	1550.62	1036.78	1539.55	1598.56	1565.82	1563.36	1531.80	1595.79	1659.45	1556.04
	Avg	716.01	472.37	718.67	733.60	720.99	723.81	707.00	723.04	721.58	709.92

4.3. Experiment 2: Comparison results on CEC2013 test suite

In the second experiment, CEC2013 test suite is applied to evaluate the performance of the peer algorithms. In the test suite, the 28 functions (F_1-F_{28}) are separated into three groups, including 5 unimodal functions (F_1-F_5), 15 multimodal functions (F_6-F_{20}), and 8 composition functions (F_{21}-F_{28}). The details CEC2013 test suite can refer to the literature [49].

All the parameters settings for this experiment are the same as the last experiment. Due to the space limitation, only the solution accuracy and t-test results and corresponding discussions are presented in this section.

4.3.1. Comparison on solution accuracy

In this section, experiments are conducted to compare the performance of each algorithm in terms of the mean and standard deviations of the solutions. The corresponding results are provided in Table 5, where the best result on each

function among all algorithms is shown in bold.

Table 5: Comparison results of solution accuracy on CEC2013 test suite.

		CLPSO	FPSO	OLPSO	SLPSO	QPSO	PSODDS	SL_PSO	HCLPSO	EPSO	FMPSO
	Mean	7.58E-14	1.44E-13	8.60E-09	5.23E-13	2.05E-13	3.57E+01	1.29E-13	3.27E-13	2.50E-13	**0.00E+00**
F_1	Std.Dev	1.01E-13	1.06E-13	1.12E-09	1.43E-13	4.09E-14	2.08E+02	1.15E-13	1.13E-13	4.09E-14	0.00E+00
	p-value	♠0.000+	♠0.000+	♠0.000+	♠0.000+	♠0.000+	♠0.000+	♠0.000+	♠0.000+	♠0.000+	
	Mean	1.58E+07	5.20E+06	4.10E+07	3.01E+06	2.41E+06	1.10E+06	5.45E+05	1.36E+06	**1.73E+05**	6.49E+06
F_2	Std.Dev	3.04E+06	1.14E+06	1.46E+07	2.09E+06	1.19E+06	2.83E+06	2.73E+05	7.10E+05	5.20E+04	5.73E+06
	p-value	♠0.008+	♠0.000-	♠0.000+	♠0.000-	♠0.000-	♠0.000-	♠0.000-	♠0.000-	♠0.000-	
	Mean	4.46E+07	6.35E+07	3.32E+10	2.51E+09	3.57E+07	4.76E+09	**2.17E+07**	3.61E+07	1.50E+08	2.64E+07
F_3	Std.Dev	2.13E+07	8.79E+07	1.48E+10	2.61E+09	2.87E+07	5.80E+09	2.14E+07	4.54E+07	1.81E+08	2.38E+07
	p-value	♠0.005-	0.051-	♠0.000+	♠0.000+	0.375+	♠0.001+	0.077-	0.170-	♠0.000+	
	Mean	1.34E+04	5.84E+03	1.31E+05	4.73E+04	6.31E+02	3.20E+03	5.95E+03	2.45E+03	**1.66E+02**	1.24E+04
F_4	Std.Dev	2.84E+03	1.25E+03	1.99E+04	1.07E+04	2.61E+02	2.76E+03	2.13E+03	1.05E+03	9.17E+01	5.27E+03
	p-value	♠0.001+	♠0.000-	♠0.000+	0.189+	♠0.000-	0.274-	♠0.000-	♠0.000-	♠0.000-	
	Mean	1.14E-13	1.14E-13	9.32E-09	4.71E-11	1.21E-13	3.54E+01	1.21E-13	4.07E-13	2.77E-13	**0.00E+00**
F_5	Std.Dev	0.00E+00	1.62E-14	6.51E-10	3.59E-10	1.41E-14	5.55E+01	2.88E-14	9.72E-14	5.91E-14	0.00E+00
	p-value	♠0.000+	♠0.000+	♠0.000+	♠0.000+	♠0.003+	♠0.000+	♠0.003+	♠0.000+	♠0.000+	
	Mean	2.61E+01	3.51E+01	4.76E+01	2.71E+01	2.18E+01	5.91E+01	1.84E+01	1.62E+01	**1.17E+01**	2.05E+01
F_6	Std.Dev	7.69E+00	2.51E+01	6.53E+00	2.12E+01	9.27E+00	3.15E+01	9.65E+00	2.96E+00	3.83E+00	1.26E+01
	p-value	0.167+	0.161+	♠0.001+	0.915+	0.069+	0.089+	0.062-	0.052-	♠0.001-	
	Mean	5.48E+01	3.53E+01	1.66E+02	9.81E+01	2.79E+01	1.08E+02	5.70E+00	**2.31E+01**	3.23E+01	3.65E+01
F_7	Std.Dev	7.44E+00	1.21E+01	3.70E+01	2.19E+01	1.22E+01	2.57E+01	4.83E+00	9.53E+00	8.99E+00	6.83E+00
	p-value	♠0.003+	0.099-	♠0.001+	0.501+	♠0.021-	0.691+	♠0.014-	0.167-	0.137-	
	Mean	**2.09E+01**	**2.09E+01**	2.10E+01	**2.09E+01**	2.10E+01	**2.09E+01**	**2.09E+01**	**2.09E+01**	**2.09E+01**	**2.09E+01**
F_8	Std.Dev	3.98E-02	4.97E-02	5.66E-02	4.88E-01	4.40E-02	4.85E-02	4.85E-02	4.52E-02	4.69E-02	3.49E-02
	p-value	0.894+	0.682+	0.425+	0.625+	0.699+	0.814+	0.894+	0.920+	0.405+	
	Mean	2.70E+01	1.54E+01	2.38E+01	3.02E+01	1.79E+01	2.60E+01	**1.01E+01**	2.03E+01	2.24E+01	2.15E+01
F_9	Std.Dev	1.23E+00	2.36E+00	2.83E+00	2.26E+00	3.93E+00	3.95E+00	2.52E+00	3.82E+00	2.76E+00	3.36E+00
	p-value	♠0.000+	0.106-	♠0.019+	0.068+	0.529-	0.081+	♠0.025-	0.907-	0.188+	
	Mean	7.43E-01	1.99E-01	1.14E+02	4.22E-01	2.20E-01	6.05E+01	2.62E-01	2.63E-01	1.53E-01	**1.02E-01**
F_{10}	Std.Dev	2.57E-01	6.18E-02	6.06E+01	2.17E-01	1.00E-01	7.73E+01	1.28E-01	1.51E-01	7.15E-02	4.02E-02
	p-value	♠0.000+	♠0.036+	♠0.000+	♠0.001+	♠0.001+	♠0.000+	♠0.006+	♠0.001+	0.099+	
	Mean	1.57E-11	5.19E+01	1.05E+00	1.52E-13	1.57E+01	7.20E+01	1.48E+01	**1.10E-13**	1.16E-13	5.74E+01
F_{11}	Std.Dev	2.73E-11	1.24E+01	3.21E+00	4.56E-14	3.41E+00	2.44E+01	4.70E+00	3.16E-14	2.59E-14	1.14E+01
	p-value	♠0.000-	0.685-	♠0.000-	♠0.000-	♠0.000-	♠0.004+	♠0.000-	♠0.000-	♠0.000-	
	Mean	1.03E+02	1.72E+02	1.37E+02	1.09E+02	7.11E+01	1.57E+02	1.62E+02	**6.05E+01**	6.79E+01	6.54E+01
F_{12}	Std.Dev	1.04E+01	7.59E+00	4.56E+01	3.23E+01	2.99E+01	4.79E+01	9.09E+00	1.84E+01	1.51E+01	1.90E+01
	p-value	♠0.014+	♠0.001+	♠0.000+	0.251+	0.104+	♠0.012+	♠0.001+	0.134-	0.254+	
	Mean	1.30E+02	1.75E+02	2.04E+02	1.74E+02	**1.07E+02**	2.58E+02	1.60E+02	1.46E+02	**1.07E+02**	1.45E+02
F_{13}	Std.Dev	1.48E+01	8.55E+00	3.05E+01	3.22E+01	2.35E+01	4.95E+01	9.67E+00	1.36E+01	2.91E+01	2.65E+01
	p-value	♠0.004-	♠0.000+	0.983+	0.880+	0.509-	♠0.007+	♠0.000+	♠0.000+	0.581-	
	Mean	5.39E+01	2.73E+03	3.85E+01	**6.32E-02**	3.97E+03	2.00E+03	7.14E+02	1.04E+01	3.17E+01	1.90E+03
F_{14}	Std.Dev	1.01E+01	5.35E+02	5.68E+01	3.00E-02	1.39E+03	6.18E+02	2.44E+02	3.23E+01	1.04E+01	4.45E+02
	p-value	♠0.000-	0.263+	♠0.000-	♠0.000-	♠0.000+	0.910+	♠0.009-	♠0.000-	♠0.000-	
	Mean	5.28E+03	6.19E+03	7.46E+03	4.28E+03	6.95E+03	4.09E+03	4.74E+03	3.97E+03	3.92E+03	**3.42E+03**
F_{15}	Std.Dev	3.34E+02	5.35E+02	1.07E+03	5.38E+02	3.05E+02	6.74E+02	2.38E+03	3.59E+02	2.08E+02	4.10E+02
	p-value	♠0.002+	♠0.001+	♠0.018+	0.472+	♠0.000+	♠0.025+	♠0.000+	0.119+	♠0.002+	
	Mean	2.44E+00	2.48E+00	3.34E+00	1.09E+00	2.50E+00	**7.03E-01**	2.47E+00	1.61E+00	1.81E+00	1.35E+00
F_{16}	Std.Dev	1.90E-01	2.01E-01	4.50E-01	2.92E-01	1.83E-01	3.63E-01	2.65E-01	1.42E-01	2.64E-01	4.33E-01
	p-value	♠0.000+	♠0.002+	0.193+	♠0.014-	♠0.000+	0.949-	♠0.000+	♠0.000+	♠0.003+	
	Mean	5.00E+01	1.70E+02	3.07E+01	**3.01E+01**	1.01E+02	1.01E+02	1.61E+02	3.06E+01	3.65E+01	1.02E+02
F_{17}	Std.Dev	2.54E+00	9.87E+00	1.93E-01	3.04E+00	3.63E+01	1.89E+01	1.32E+01	9.62E-02	2.32E+00	2.83E+01
	p-value	♠0.000-	♠0.000+	♠0.000-	♠0.000-	0.146-	♠0.004+	♠0.001+	♠0.000-	♠0.000+	
	Mean	2.03E+02	1.98E+02	2.29E+02	1.44E+02	1.97E+02	1.59E+02	1.92E+02	9.74E+01	8.68E+01	**8.02E+01**
F_{18}	Std.Dev	1.04E+01	7.37E+00	2.54E+01	2.89E+01	1.20E+01	3.74E+01	9.93E+00	2.70E+01	1.30E+01	1.25E+01
	p-value	0.308+	♠0.009+	0.366+	♠0.015+	0.762+	♠0.008+	♠0.010+	♠0.001+	0.850+	
	Mean	3.74E+00	1.21E+01	2.54E+00	**1.02E+00**	3.38E+00	8.08E+00	3.43E+00	1.42E+00	1.84E+00	5.10E+00
F_{19}	Std.Dev	3.29E-01	7.81E-01	4.65E-01	3.09E-01	9.31E-01	7.02E+00	5.25E-01	2.45E-01	2.73E-01	1.18E+00
	p-value	♠0.000-	♠0.047+	♠0.000-	♠0.000-	0.430-	♠0.000+	♠0.000-	♠0.000-	♠0.000-	
	Mean	1.31E+01	1.30E+01	1.44E+01	1.26E+01	1.19E+01	1.42E+01	1.38E+01	**1.10E+01**	**1.10E+01**	1.27E+01
F_{20}	Std.Dev	3.90E-01	1.13E+00	5.89E-01	8.12E-01	4.60E-01	1.01E+00	1.34E+00	7.39E-01	6.97E-01	4.78E-01
	p-value	0.396+	0.088+	0.730+	0.132-	0.848-	♠0.013+	♠0.000+	0.668-	0.118-	
	Mean	2.80E+02	3.09E+02	**2.19E+02**	2.96E+02	2.92E+02	3.00E+02	2.97E+02	2.28E+02	2.26E+02	2.40E+02
F_{21}	Std.Dev	4.17E+01	3.58E+01	5.13E+01	8.21E+01	6.80E+01	9.53E+01	8.15E+01	4.18E+01	4.23E+01	4.80E+01
	p-value	0.241+	0.126+	0.281-	0.108+	0.088+	♠0.004+	♠0.002+	0.744+	0.310-	
	Mean	1.96E+02	1.70E+03	3.00E+02	**1.08E+02**	1.98E+03	2.32E+03	6.05E+02	1.12E+02	1.48E+02	1.31E+02
F_{22}	Std.Dev	3.51E+01	6.48E+02	1.83E+02	6.38E+01	1.07E+03	6.37E+02	2.64E+02	1.60E+01	2.28E+01	3.65E+01
	p-value	0.835+	♠0.000+	♠0.000+	0.310-	♠0.000+	♠0.000+	♠0.000+	♠0.000-	0.051+	
	Mean	5.86E+03	6.79E+03	7.47E+03	4.93E+03	7.07E+03	4.87E+03	4.26E+03	4.11E+03	4.00E+03	**3.98E+03**
F_{23}	Std.Dev	3.48E+02	3.43E+02	1.24E+03	6.53E+02	3.30E+02	9.48E+02	2.50E+03	5.31E+02	5.46E+02	6.79E+02
	p-value	♠0.001+	♠0.000+	♠0.001+	0.169+	♠0.000+	0.774+	♠0.000+	♠0.003+	0.143+	
	Mean	2.58E+02	2.48E+02	2.73E+02	2.75E+02	2.44E+02	2.72E+02	2.44E+02	**2.30E+02**	2.47E+02	2.87E+02
F_{24}	Std.Dev	7.72E+00	1.99E+01	5.44E+00	9.58E+00	5.22E+00	1.03E+01	1.18E+01	8.21E+00	8.14E+00	7.74E+00
	p-value	0.986-	♠0.000-	♠0.016-	0.717-	0.090-	0.424-	0.922-	0.180-	0.759-	
	Mean	2.89E+02	2.62E+02	2.83E+02	2.93E+02	2.60E+02	2.96E+02	**2.53E+02**	2.71E+02	2.84E+02	2.98E+02
F_{25}	Std.Dev	3.55E+00	2.22E+01	5.49E+00	7.96E+00	5.46E+00	1.10E+01	5.93E+00	1.84E+01	7.04E+00	7.12E+00
	p-value	♠0.002-	♠0.000-	♠0.039-	0.686-	0.176-	0.638+	♠0.018-	0.131-	0.912-	
	Mean	2.01E+02	3.12E+02	2.28E+02	2.32E+02	2.88E+02	2.41E+02	2.44E+02	**2.00E+02**	2.08E+02	3.45E+02
F_{26}	Std.Dev	3.36E-01	3.73E+01	5.35E+01	6.74E+01	6.47E+01	7.00E+01	5.35E+01	2.69E-02	1.56E+01	2.97E+01
	p-value	♠0.000-	0.432-	0.185-	♠0.016-	♠0.000-	♠0.001-	♠0.013-	♠0.000-	0.125-	
	Mean	8.14E+02	6.06E+02	9.44E+02	1.07E+03	7.23E+02	9.60E+02	**4.84E+02**	5.95E+02	7.27E+02	6.04E+02
F_{27}	Std.Dev	2.32E+02	8.28E+01	5.96E+02	7.91E+01	6.42E+01	1.07E+02	1.21E+02	1.58E+02	1.48E+02	2.37E+02
	p-value	0.804+	♠0.000-	♠0.000+	♠0.000+	♠0.000+	♠0.000+	0.862-	0.067-	♠0.000+	
	Mean	**3.00E+02**	3.10E+02	1.26E+03	5.78E+02	4.07E+02	1.05E+03	3.11E+02	3.14E+02	**3.00E+02**	**3.00E+02**
F_{28}	Std.Dev	8.35E-11	2.77E+02	2.63E+02	5.67E+02	1.92E+02	6.36E+02	1.05E-05	1.06E+02	6.50E-10	8.03E-11
	p-value	0.773+	♠0.000-	♠0.000+	♠0.004+	♠0.000+	♠0.000+	♠0.043+	♠0.036+	0.494+	
(#)	Best-Mean	2	1	1	5	1	2	5	7	7	8
(#)	♠+	11	12	15	7	10	16	14	8	7	
(#)	♠ -	8	6	6	7	5	2	9	8	6	

From Table 5 we can see that FMPSO dominates other 9 peer algorithms in terms of the number of obtained best mean values on the 28 functions, followed by EPSO and HCLPSO, both of which offer the best results on 7 test function. The performances of the peer algorithms on the 3 different types of problems are detailed as follows.

1) *Unimodal Functions* (F_1-F_5): On the 5 unimodal functions, FMPSO and EPSO manifest the most favorable performance since they both achieve the best solutions on 2 out of the 5 problems, followed by SL_PSO who obtains the best result on one function. It is worth to note that FMPSO is the only algorithm that attains the global optimal solutions on both F_1 and F_5 on all runs. Although all the 10 peer algorithms cannot find out the global optimal solutions on those non-separable unimodal problems, i.e., F_2, F_3, and F_4, EPSO displays more competitive performance since it offers the best results on 2 out of the 3 non-separable problems.

2) *Basic Multimodal Functions* (F_6-F_{20}): In this type functions, HCLPSO displays the most promising performance on 5 out of the 15 multimodal functions, followed by FMPSO and EPSO, both of which yield the best mean solutions on 4 multimodal functions. Since the majority of the 15 functions are non-separable problems, no algorithm find out the global optimal solutions of these problems. Although CLPSO is the common component of HCLPSO and EPSO, the latter two algorithms manifest more superior performance. Considering that the common feature between HCLPSO and FMPSO is the multi-swarm structure, we can obtain a conservative conclusion that the multi-swarm mechanism as well as multi-role for particles play positive performance on multimodal problems.

3) *Composition Functions* (F_{21}-F_{28}): For the 8 composition functions, FMPSO, HCLPSO, and SL_PSO display the same performance, in terms of the number of the best mean values achieved by them. Although the 3 outstanding algorithms present more superior performance than other peer algorithms, the advantages of them on the composition functions is very small for all the peer algorithms have been trapped into local optimal in these functions.

The comparison results on the composition functions demonstrate that the performance of FMPSO on the complicated problems need further improvement. Thus, we regard that a crucial issue must be deal with is that how to draw much essential characteristics from the problems by the population's historical experience.

4.3.2. t-test results

Table 5 shows that FMPSO and HCLPSO have the same performance because the numbers of the problems they dominate each other are both 8 although FMPSO yields the best mean results on 8 functions which is slightly better than that of HCLPSO. Furthermore, SLPSO also displays the same performance as FMPSO, in terms of the t-test results.

In the unimodal problems, four outstanding algorithms, i.e., FMPSO, HCLPSO, EPSO and SL_PSO, manifest almost the same performance. However, SLPSO offers the most favorable results since it outperforms FMPSO on 5 out of the 15 multimodal functions while it dominated by FMPSO only on 2 functions. Furthermore, EPSO and HCLPSO show the same performance, in terms of the t-test results. Although SL_PSO has the same performance as FMPSO, it displays very unfavorable results on the multimodal problems. On the 8 composition functions, although FMPSO, HCLPSO, and SL_PSO display more pleasurable performance, in terms of the number of achieved best mean results, F_PSO shows the most outstanding results since it dominates FMPSO on 4 out of the 8 composition functions. Furthermore, HCLPSO demonstrates almost the same performance as FMPSO on this type functions. Although different peer algorithms give various results on different composition problems, none of them can find out a global optimum in each one problem.

4.4. Experiment 3: Sensitivity analysis of components

There are four key additional strategies in FMPSO: the multi-role of particles, the "global tuning", the "local tuning", and the "local searching" operators. Considering that the relationship among the new introduced parameters (i.e.,

c_1^i, c_2^i, c_3^i, s_{d1}^i, s_{d2}^i, and s_{d3}^i) may be very complex and correlated, we only analysis the sensitivity of the "global tuning", the "local tuning", and the "local searching" operators. Due to the space limitation, we carry out a series of experiments only on the f_1 to f_{14} to find out how the three components affect the performance of FMPSO. The experimental results are listed in Table 6, in which FMPSO-GT and FMPSO-LT indicate the modified algorithms that removing the "global tuning" and "local tuning" components from FMPSO, respectively. Moreover, to find out whether carrying out the local searching at the last evolutionary stage is the most favorable choice, we replace the local searching in FMPSO by a periodical local searching operator. For simplicity, the FMPSO variant is called as FMPSO-LS, in which the local searching operator is carried out while the "global tuning" components is conducted. In other words, **Gb** will be refined by the local searching operator when it stagnate.

The comparison results listed in Table 6 demonstrate that FMPSO dominates FMPSO-LT on all the test functions. The results indicate that "local tuning" component plays a critical role not only on unimodal functions but also on multimodal problems. Furthermore, the results between FMPSO and FMPSO-GT also verify the positive performance of the "global tuning" component. However, the active impaction of the "global tuning" is smaller than that of the "local tuning". The reason is that the "local tuning" has a higher speed of information diffusion which is very important for swarm intelligence algorithms.

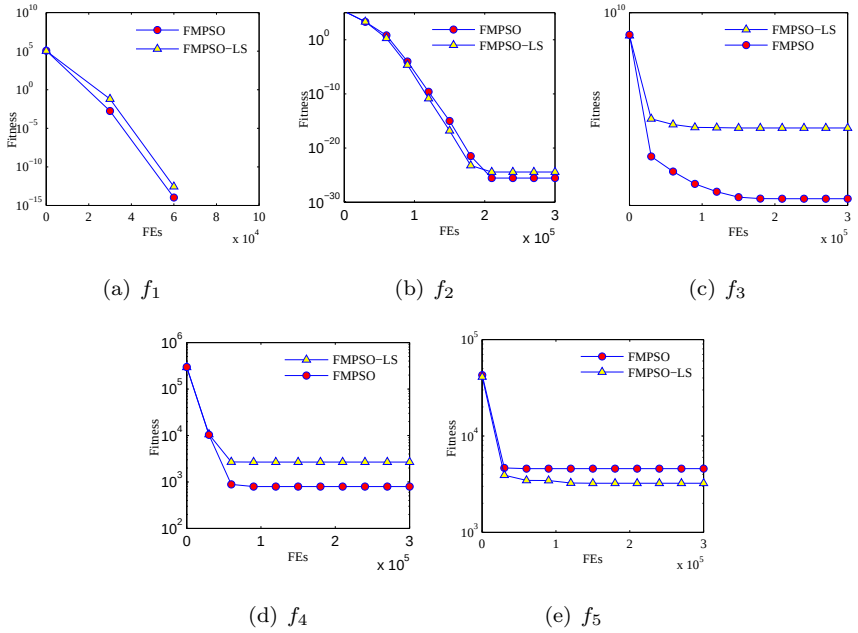
From the results offered by FMPSO and FMPSO-LS, we can see that difference of the performance between them are not significant than that between FMPSO and other two algorithms. In addition, FMPSO-LS dominates FMPSO on some problems. Since the effect of the local searching component is to refine **Gb**, and then to endow FMPSO with a high convergence speed, the convergence processe of FMPSO and FMPSO-LS attract much attention of us. The comparison results of convergence progresses on the unimodal functions and multimodal functions between the two algorithm are listed in Fig. 5 and Fig. 6, respectively.

Table 6: Comparison results of components on f_1 to f_{14}.

		FMPSO-GT	FMPSO-LT	FMPSO-LS	FMPSO
f_1	Mean	**0.00E+00**	9.01E+03	**0.00E+00**	**0.00E+00**
	Std.Dev	0.00E+00	4.15E+03	0.00E+00	0.00E+00
f_2	Mean	4.40E-18	4.70E+04	1.06E-22	**4.37E-25**
	Std.Dev	8.34E-18	1.11E+04	1.95E-22	4.65E-25
f_3	Mean	5.47E+06	2.05E+08	5.53E+06	**1.10E+06**
	Std.Dev	5.28E+06	5.91E+07	4.60E+06	7.34E+05
f_4	Mean	1.98E+03	5.97E+04	2.30E+03	**1.43E+03**
	Std.Dev	1.47E+03	1.59E+04	1.16E+03	6.75E+02
f_5	Mean	4.84E+03	1.28E+04	**4.01E+03**	4.17E+03
	Std.Dev	1.03E+03	1.95E+03	1.08E+03	1.25E+03
f_6	Mean	2.30E+00	1.66E+09	**6.64E-01**	9.44E-01
	Std.Dev	2.61E+00	1.11E+09	1.11E+00	1.42E+00
f_7	Mean	2.24E-02	3.75E+02	2.42E-02	**1.29E-02**
	Std.Dev	1.21E-02	1.60E+02	1.86E-02	9.41E-03
f_8	Mean	2.03E+01	2.10E+01	2.02E+01	**1.97E+01**
	Std.Dev	3.23E-01	5.72E-02	2.39E-01	1.31E+00
f_9	Mean	3.90E+01	1.28E+02	3.99E+01	**3.50E+01**
	Std.Dev	9.04E+00	2.50E+01	6.72E+00	7.50E+00
f_{10}	Mean	6.24E+01	3.12E+02	6.28E+01	**4.75E+01**
	Std.Dev	1.23E+01	3.19E+01	1.47E+01	1.01E+01
f_{11}	Mean	2.06E+01	4.08E+01	2.01E+01	**1.90E+01**
	Std.Dev	3.00E+00	1.72E+00	4.14E+00	2.69E+00
f_{12}	Mean	4.76E+04	3.38E+05	4.38E+04	**2.12E+04**
	Std.Dev	3.67E+04	9.05E+04	3.58E+04	1.65E+04
f_{13}	Mean	3.78E+00	5.80E+01	3.99E+00	**2.74E+00**
	Std.Dev	9.84E-01	1.82E+01	7.51E-01	8.13E-01
f_{14}	Mean	1.23E+01	1.37E+01	1.21E+01	**1.19E+01**
	Std.Dev	4.07E-01	1.50E-01	4.28E-01	4.24E-01

FMPSO-GT and FMPSO-LT represent two FMPSO variants in which the "global tuning" and "local tuning" components are removed from FMPSO, respectively. FMPSO-LS denotes a FMPSO variant in which the local searching component is replaced by a periodically searching operator.

From Fig. 5 we can see that FMPSO attains a higher convergence speed than FMPSO-LS on f_1, f_3, and f_4 during the entire evolutionary progress. On the contrary, FMPSO-LS yields better performance on f_5, in terms of the convergence speed. Furthermore, FMPSO-LS shows higher convergence speed on f_2 during the initial evolutionary stage while FMPSO offers more accurate solution of f_2 benefited from its higher convergence speed at the later optimization stage. Fig. 6 indicates that FMPSO is significant better than FMPSO-LS on f_{10}, f_{12}, and f_{14}, in terms of the convergence speed. On the contrary, FMPSO-LS demonstrates more favorable performance on f_6. At other 5 multimodal functions, the convergence speeds of the two algorithms are not significant dif-



(a) f_1 (b) f_2 (c) f_3

(d) f_4 (e) f_5

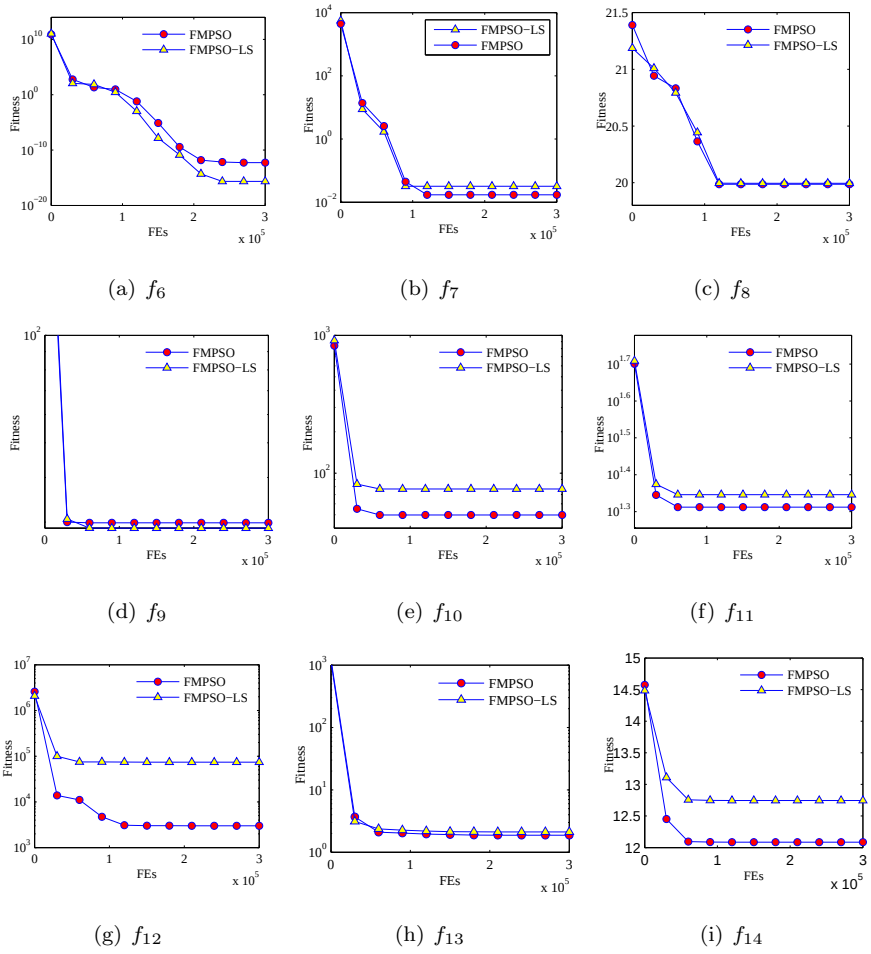
Figure 5: Convergence progress of FMPSO and FMPSO-LS on the 5 unimodal functions.

ference.

Although we have a rudimentary understanding of how the different components affect the performance of FMPSO, the configurations of the components may not be the best settings for a particular problem because there may be dependency between the different components, which causes the combination of each component no long to be an optimal setting for FMPSO.

5. Conclusion

This paper presents a variant of PSO which named by us as a fitness-based multi-role PSO, FMPSO for short. In FMPSO, the update rule of velocity is expended from 3 parts in the canonical PSO to 4 parts including inertia part, self-cognitive part, social-learning part, and subsocial-learning part. A particle's exemplars of the later three parts are its historical best position, the



(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}

Figure 6: Convergence progress of FMPSO and FMPSO-LS on the 9 multimodal functions.

population's best position, and its neighbors' best position, respectively. The main advantage of this modification is taking full advantages of *gbest* and *lbest* topologies.

At each generation, different particles play different roles, including leader, rambler, and follower, based on their fitness. Accordingly, particles that playing different roles will be assigned different learning weights for the three parts,

i.e., self-cognitive, social-learning, and subsocial-learning, the aim of which is to help the population carry out various search mechanisms. Moreover, during the evolutionary process, the roles assigned for different particles can be adjusted by the two tuning operators. Accordingly, the particles reselect their own learning model. Specifically, when the population's historical best solution has stagnated, all particles adjust their learning exemplars and objective dimension which called "global tuning". In another case that some individuals have stagnated while the global best solution has not suspended improvement, only those stagnated individuals reselect their objective dimensions, which named as "local tuning" in this research.

Experimental results show that the proposed FMPSO outperforms the other state-of-art PSO variants on a majority of the test functions in terms of the global search ability, solution accuracy, and convergence speed. Furthermore, the results of two statistical tests also verify the promising performance of FMPSO. At last, experiments aiming to analysis the sensitivity and effectiveness of the main components in FMPSO are also conducted. Although a rudimentary understanding of components' performance has been obtained, there are many issues need to be further studied in our future works.

However, we must admit that when to carry out the global tuning and local tuning strategies for a specific problem needs to be further studied. For example, conducting the tuning strategies too frequently may disturb the current favorable search direction of population and individuals. On the contrary, if the population and individuals do not timely perform the tuning strategies while they have been stagnated, the current search model may waste too many computing resource. Thus, we regard that it is beneficial for improving FMPSO performance that extracting much useful information from characteristics of the current fitness landscape. We will pay more attention on it in our future works.

Acknowledgment. This study was funded by the National Natural Science Foundation of China (Grant Nos.: 61663009, 61602174), the National Natural Science Foundation of Jiangxi Province (Grant Nos.: 20171BAB202019, 20161BAB202064, 20161BAB212052, 20151BAB207022), the National Natu-

ral Science Foundation of Jiangxi Provincial Department of Education (Grant Nos.: GJJ160469), and the Research Project of Jiangxi Provincial Department of Communication and Transportation (No. 2017D0038).

Reference

[1] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of IEEE international conference on neural network, 1995, pp.1942-1948.

[2] J. Kennedy, R.C. Eberhart, A new optimizer using particle swarm theory, in: Proceedings of the sixth international symposium on Micro machine and Human Science, 1995, pp.39-43.

[3] S. He, E. Prempain , Q. Wu, An improved particle swarm optimizer for mechanical design optimization problems, Eng. Optim. 36(5)(2004) 585-605.

[4] D. Sha, C.Y. Hsu, A new particle swarm optimization for the open shop scheduling problem, Comput. Oper. Res. 35(10)(2008) 3243-3261.

[5] M. Gong, Q. Cai, X. Cheng, *et al.*, Complex network Clustering by multiobjective discrete particle swarm optimization based on decomposition, IEEE Trans. Evol. Comput. 18(3)(2014) 82-97.

[6] G. Pedram, S. Micael, M.L. Fernando, *et al.*, Multilevel image segmentation based on fractional-order darwinian particle swarm optimization, IEEE Trans. Geos. Remote Sensing. 52(5)(2014) 2382-2394.

[7] J. Kennedy, Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance, in: Proceedings IEEE Congress on evolutionary computation, 1999, pp. 1931-1938.

[8] M. Clerc, J. Kennedy, The particle swarm - explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evol. Comput. 6(2)(2002) 58-73.

[9] S. Janson, M. Middendorf, A hierarchical particle swarm optimizer and its adaptive variant, IEEE Trans. Syst., Man, Cybern. B, Cybern. 35(6)(2005) 1272-1282.

[10] V. Kadirkamanathan, K. Selvarajah, P.J. Fleming, Stability analysis of the particle dynamics in particle swarm optimizer, IEEE Trans. Evol. Comput. 10(3)(2006) 245-255.

[11] Z.H. Zhan, J. Zhang, Y. Li *et al.*, Adaptive particle swarm optimization. IEEE Trans. Syst., Man, Cybern. B, Cybern. 39(6)(2009) 1362-1381.

[12] Y.J. Zheng, H.F. Ling, Q. Guan, Adaptive parameters for a modified comprehensive learning particle swarm optimizer. Math. Probl. Eng. 68(3)(2009) 939-955.

[13] G. Xu, An adaptive parameter tuning of particle swarm optimiztion algorithm. Appl. Math. Comput. 219(9)(2009) 4560-4569.

[14] R. Eberhart, Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings IEEE Congress on evolutionary computation, 2000, pp. 84-88.

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

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

[17] J.J. Liang, P.N. Suganthan, Dynamic multi-swarm particle swarm optimizer. In: Proceedings of IEEE symposium on swarm intelligence, 2005, pp. 124-129.

[18] S.Z. Zhao, J.J. Liang, P.N. Suganthan, *et al.*, Dynamic multi-swarm particle optimizer with local search for large scale global optimization. In: Proceedings of IEEE congress on evolutionary computation, 2008, pp. 3845-3852.

[19] Hüseyin Haklı, Harun Uğuz, A novel particle swarm optimization algorithm with Levy flight. Appl. Soft Comput. 23(5)(2014) 333-345.

[20] S. Sayah, A. Hamouda, A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex ecomomic dispatch problems, Appl. Soft Comput. 13(4)(2013) 1608-1619.

[21] A. Sedki, D. Ouazar, Hybrid particle swarm optimization and differential evolution for optimal design of water distribution systems, Adv. Eng. Inform. 26(3)(2012) 582-591.

[22] Y. Shi, R.C. Eberhart, Fuzzy adaptive particle swarm optimization, in: Proceedings of IEEE congress on evolutionary computation, 2001, pp. 101-106.

[23] A. Chatterjee, P. Siarry, Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization, Comput. Oper. Res. 33(3)(2004) 859-871.

[24] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: Proceedings of IEEE world congress on computational intelligence, 1998, pp. 68-73.

[25] 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.

[26] M.R. Tanweer, s. Suresh, N. Sundararajan, Self regulating particle swarm optimization algorithm, Inf. Sci. 294(2015) 182-202.

[27] G. Karafotias, M. Hoogendoorn, A.E. Eiben, Parameter control in evolutionary algorithms: trends and challenges, IEEE Trans. Evol. Comput. 19(2)(2015) 167-187.

[28] Kennedy J., Mendes R., Population structure and particle swarm performance, in: Proceedings of IEEE congress on evolutionary computation, 2002, pp.1671-1676.

[29] P.N. Suganthan, Particle swarm optimiser with neighbourhood operator, in: Proceedings of IEEE congress on evolutionary computation, 1999, pp.1958-1962.

[30] T. Peram, K. Veeramachaneni, C.K. Mohan, Fitness-distance-ratio based particle swarm optimization, in: Proceedings of IEEE symposium on swarm intelligence, 2003, pp.174-181.

[31] 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. 10(3)(2006) 281-295.

[32] J. Sun, W. Fang, X.J. Wu *et al.*, Quantum-behavior Particle swarm optimization: Analysis of individual particle behavior and parameter selection, Evol. Comput. 20(3)(2012) 349-393.

[33] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm Evol. Comput. 24(2015) 11-24.

[34] R. Cheng, Y.C. Jin, A social learning particle swarm optimization algorithm for scalable optimization, Inf. Sci. 291(6)(2015) 43-60.

[35] J. Jie, W.L. Wang, C.S. Liu, *et al.*, Multi-swarm Particle Swarm Optimization Based on Mixed Search Behavior, in: Proceedings of fifth IEEE Conference on industrial electronics and applications, 2010 pp. 605-610.

[36] X.D. Li, Niching without niching parameters: particle swarm optimization using a ring topology, IEEE Trans. Evol. Comput. 14(1)(2010) 150-169.

[37] Y. Jiang, X.Y. Li, C.C. Huang, Automatic calibration a hydrological model using a master-slave swarms shuffling evolution algorithm based on self-adaptive particle swarm optimization, Expert Syst. Appl. 40(40)(2013) 752-757.

[38] P.J. Angeline, Using selection to improve particle swarm optimization, in: Proceedings of IEEE congress on evolutionary computation, 1988, pp.84-89.

[39] H. Higashi, H. Iba, Particle swarm optimization with Gaussian mutation, in: Proceedings of IEEE symposium on swarm intelligence, 2003, pp.72-79.

[40] C. Wei, Z. He, Y. Zhang *et al.*, Swarm directions embedded in fast evolutionary programming, in: Proceedings of IEEE congress on evolutionary computation, 2002, pp.1278-1283.

[41] B. Xin, J. Chen, J. Zhang, H. Fang *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. 42(5)(2012) 744-767.

[42] S. Li, M. Tan, I.W. Tsang, *et al.*, A hybrid PSO-BFGS strategy for global optimization on multimodal functions, IEEE Trans. Syst., Man, Cybern. B, Cybern. 41(4)(2011) 1003-1014.

[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. 23(5)(2014) 76-90.

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

[45] X. Jin, Y.Q. Liang, D.P. Tian, *et al.*, Particle swarm optimization using dimension selection methods, Appl. Math. Comput. 219(10)(2013) 5185-5197.

[46] N. Lynn, P.N. Suganthan, Ensemble particle swarm optimizer, Appl. Soft Comput. 55(2017) 533-548.

[47] X.W. Xia, C.W. Xie, B. Wei, Z.B. Hu, B.J. Wang, C. Jin, Particle swarm optimization using multi-level adaptation and purposeful detection operators, Inf. Sci. 385(2017) 174-195.

[48] P.N. Suganthan, N. Hansen, J.J. Liang, *et al*, Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. KanGAL Report 2005005, 2005.

[49] 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.