Mathematical Biosciences
and Engineering

http://www.aimspress.com/journal/mbe

MBE, 20(4): 7056–7087.
DOI: 10.3934/mbe.2023305
Received: 10 December 2022
Revised: 11 January 2023
Accepted: 16 January 2023
Published: 09 February 2023

Research article

A novel particle swarm optimization based on hybrid-learning model

Yufeng Wang[1], **BoCheng Wang**[1,*], **Zhuang Li**[1] **and Chunyu Xu**[2,3]

[1] School of Computer Science and Software, Nanyang Institute of Technology, Nanyang 473306, China

[2] Electronic Information School, Wuhan University, Wuhan 430072, China

[3] School of Information Engineering, Nanyang Institute of Technology, Nanyang 473306, China

* **Correspondence:** Email: bochengwang@nyist.edu.cn.

Abstract: The convergence speed and the diversity of the population plays a critical role in the performance of particle swarm optimization (PSO). In order to balance the trade-off between exploration and exploitation, a novel particle swarm optimization based on the hybrid learning model (PSO-HLM) is proposed. In the early iteration stage, PSO-HLM updates the velocity of the particle based on the hybrid learning model, which can improve the convergence speed. At the end of the iteration, PSO-HLM employs a multi-pools fusion strategy to mutate the newly generated particles, which can expand the population diversity, thus avoid PSO-HLM falling into a local optima. In order to understand the strengths and weaknesses of PSO-HLM, several experiments are carried out on 30 benchmark functions. Experimental results show that the performance of PSO-HLM is better than other the-state-of-the-art algorithms.

Keywords: particle swarm optimization; adaptive parameters; multi-pool fusion; hybrid-learning model; Gaussian Perturbation

1. Introduction

Meta-heuristic algorithm is a method based on computational intelligence to solve the optimal solution of complex optimization problems. It refines the corresponding feature model under the guidance of the characteristics of specific problems and designs intelligent iterative search optimization algorithm through the understanding of relevant behaviors, functions, experiences, rules and action mechanisms in biological, physical, chemical, social, artistic and other systems or fields, such as the monarch butterfly optimization (MBO) [1], slime mould algorithm (SMA) [2], moth search algorithm (MSA) [3], hunger games search (HGS) [4], Runge Kutta method (RUN) [5], colony predation algorithm (CPA) [6], weIghted meaN oF vectOrs (INFO) [7] and Harris hawks optimization

(HHO) [8].

PSO is one of the important branches of meta heuristic algorithm, it was proposed by Kennedy and Eberhart in 1995 [9]. It is a heuristic swarm intelligence algorithm, which can solve global optimization problems by simulating the foraging behavior of birds. In the process of problem solving, PSO searches the global optimal solution by sharing information among particles. It is widely concerned because of its simplicity and fast convergence [10]. In order to solve a series of optimization problems [11, 12], researchers have proposed many PSO variants. By adaptively adjusting the inertia weight of the PSO algorithm [13], the convergence speed of the algorithm is accelerated. By designing different types of topology, the particle learning model is changed, which increases the diversity of the population [14]. By simulating dynamic balance and dynamically forming an independent search space, PSO can effectively avoid premature convergence [15, 16]. In addition, Levy flight is a random walk with a heavy tailed probability distribution of step size, that is, there is a relatively high probability of large strides in the process of random walking. Compared with random walking without heavy tails in the step size distribution, the motion track of Levy flight is just like flying. It can jump out of the local optimal value with high probability [17].

Three archives particle swarm optimization algorithm (TAPSO) [18] is proposed to solve continuous domain global optimization problems. This method increases the diversity of the population by increasing the evaluation criteria of the improvement rate, which improves the accuracy of the multimodal problem and optimizes the potential, but is prone to premature convergence [19]. By increasing the improvement rate, the algorithm improves the diversity of the population and the precision and optimization potential of the multimodal problem. This is a considerable improvement direction, but due to the unclear particle objectives at each stage and insufficient division of the learning model, it is easy to lead to premature convergence and sharp decline in population diversity.

In order to improve the drawbacks of premature convergence and population diversity, the population of PSO-HLM is divided into four sub-population pools (elite pool, potential pool, triple potential pool and fusion pool). The new offspring randomly select two sub-population pools for cross fusion. In the particle learning stage, the velocity of each particle can be updated through four learning models: confidence learning model, mild learning model, standard learning model, Gaussian learning model. Various learning models can guarantee that each particle information can be fully used, so that it does not fall into the local optima prematurely. In the process of iteration, PSO-HLM uses tangent function to conduct large-scale random disturbance to the inertia weight, thus it can realize large-scale particle searches. Through the adaptive change of the cross probability using the sine function, PSO-HLM mainly focuses on particles with good fitness value at the beginning of iteration. At the later stage of the iteration, it focuses on particles with high fitness value improvement rate.

The rest of this paper is organized as follows: Section 1 is an introduction. Section 2 introduces the traditional particle swarm optimization algorithm, and analyzes some variations of particle swarm optimization algorithm. Section 3 provides a detailed analysis and explanation of the PSO-HLM. Section 4 is the experimental part, which verify the feasibility and effectiveness of the PSO-HLM. Section 5 summarizes the main points of this paper and introduce the future work.

2. Related works

2.1. Canonical PSO

Particle swarm optimization (PSO) has recently developed into an important branch of combinatorial heuristic technology. It operates on a group of potential solutions to explore the optimal search space. Like ant colony optimization (ACO) [20] and other algorithms, particle swarm optimization was originally developed to study the concept of interesting social behavior (flocks of birds or fish) [21] in a simplified way in simulation. However, the potential of this technology is soon realized as a powerful optimization tool [22], which can be successfully applied to continuous and discrete optimization problems [23]. In cloud task scheduling, hybrid particle swarm optimization (HPSO) uses intelligent fish swarm algorithm and particle swarm optimization algorithm to further reduce the total execution application time [24]. Traveling Salesman Problem (TSP) is regarded as an NP hard problem, which is the benchmark of various optimization methods [25]. PSO is used to optimize the parameters of Ant Colony Optimization (ACO), and then combines with ACO to solve the TSP problem [26].

PSO is a population-based intelligent optimization algorithm, its flowchart is shown in Figure 1. The canonical PSO could be viewed as a population size n of particles searching in problem dimension d. Through continuous iterative searching and communicating, as well as the intra-group communication mechanism, the optimal value of the problem space was finally achieved. Each particle is a feasible solution of the problem, and they all have two vectors of velocity and position. The position and velocity of the *i-th* particle in each dimension at iteration time t were $x_i^t = [x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,d}^t]$ and $v_i^t = [v_{i,1}^t, v_{i,2}^t, \ldots, v_{i,d}^t]$. At each iteration, the position and velocity of the particles were updated by Eq. 2.1 and Eq. 2.2.

$$\mathrm{v}_{i,j}^t = w \cdot v_{i,j}^{t-1} + c_1 \cdot r_1 \cdot \left(pb_{i,j}^t - x_{i,j}^{t-1}\right) + c_2 \cdot r_2 \cdot \left(gb_j^t - x_{i,j}^{t-1}\right) \tag{2.1}$$

$$x_{i,j}^t = x_{i,j}^{t-1} + v_{i,j}^t \tag{2.2}$$

Where $i \in [1,n]$ is the number of particles, $j \in [1,d]$ is the dimension. pb stands for the best position of the individual particle, gb is the position of the best particle in the whole population. w is the inertia weight, c_1 and c_2 are the acceleration constants, usually equal to 2. $r_1, r_2 \in [0,1]$ are random numbers.

2.2. Studies of PSO

In the past ten years, many variants and improvements have been made to the basic PSO algorithm to deal with premature convergence and obtain higher search speed. Since the parameters used in PSO are considered to have a great impact on the performance of the algorithm, many studies have been carried out to adaptively adjust the parameters of different problems [27]. Different from the traditional particle swarm with the same particle, some researches have proposed heterogeneous particle swarm optimization algorithm, in which the particles have different behaviors. The information transmission of several neighborhood typologies is analyzed theoretically, and a PSO [28] with changing topology is proposed. In order to avoid premature convergence, a method to maintain group diversity is proposed [29]. The evolution process of PSO is deeply analyzed [30], several methods to improve the performance of PSO are described.

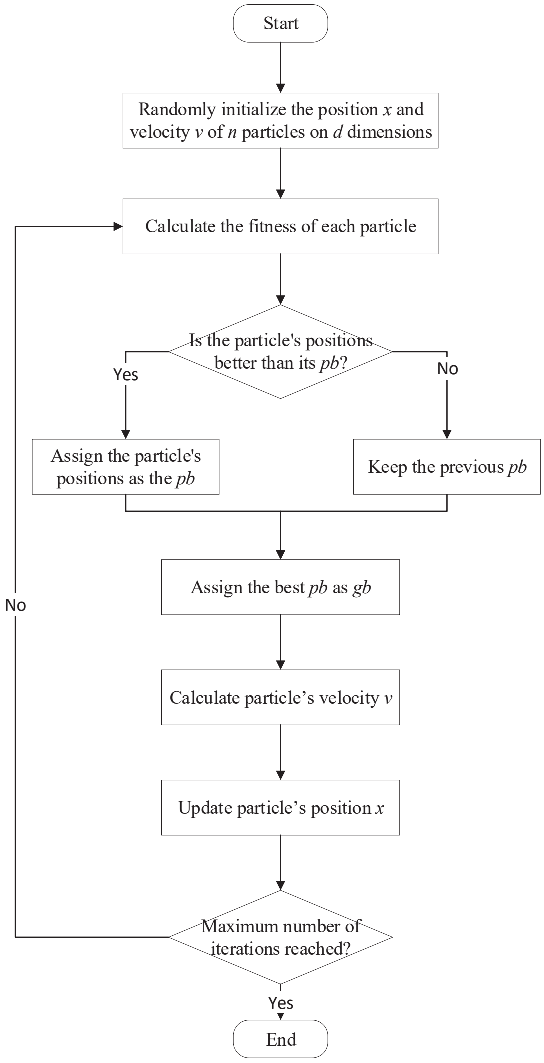

Figure 1. The flowchart of Canonical PSO.

On the adjustment and improvement of many strategies, designing a more effective speed update rule has attracted the attention of many researchers. Many scholars have proposed various PSO variants. According to the goal to be achieved, most researches can be divided into four categories, namely parameter adjustment, learning strategy adjustment and auxiliary combination with other algorithms.

1) Parameter adjustment: it is generally believed that a smaller w is conducive to exploitation, while a larger w is conducive to exploration. In the optimization process, the most common update rule of w decreases linearly from 0.9 to 0.4, which is still applied in many PSO. Based on the effect of iteration w, Rantnaweera et al. further advocate the use of layered PSO with variable time acceleration coefficient HPSO-TVAC [31, 32]. However, considering that the PSO search process is nonlinear and complex, many nonlinear change rates are proposed to adjust parameters to give particles different search behaviors. The setting of parameters will encounter inappropriate contributions to the adjustment of parameters. In order to achieve more satisfactory results in the

adjustment of parameters, many scholars propose different adjustments [22, 26]. For example, in adaptive PSO (APSO), w, c_1 and c_2 no longer depend on the number of iterations. Various experiments aim to adjust the parameters of particles by combining or separating fitness value, velocity and diversity of population. A large number of experimental data show that adaptive can help particles achieve the balance between exploration and development capabilities.

2) Learning strategy adjustment: in the research of particle swarm optimization, global PSO (GPSO) and local PSO (LPSO) are two basic learning strategies [33]. Many studies show that the sparse and dense neighbor topology [13, 14] strategies are suitable for complex multimodal problems and simple unimodal problems. In order to overcome the shortcomings of single learning, researchers use different weighting methods to take the entire population as an example of particle swarm. In this way, the effects of comprehensive learning strategy, orthogonal learning strategy, interactive learning strategy and dimensional learning strategy are significant.

3) Combination of auxiliary particles: Considering that different operators or optimization algorithms have their own characteristics, many researchers focus on combining them with reasonable integration strategies [34]. For example, genetic operators and various local search strategies are very popular auxiliary tools for balancing exploration and development [35]. In addition, levy flight, as a common random allocation strategy, is another auxiliary type of PSO [36].

4) Fully-adaptive PSO variants: some PSO variants can make usage of their cognitive and social factors knowledge learned from their neighbors, Lynn and Suganthan proposed an integrated PSO (EPSO) [12], in which five PSO variants are hybridized together through the integrated method. Experiments show that the adaptive mechanism used to allocate appropriate PSO variants to the population in the evolutionary process can organically integrate the advantages of the PSO variables involved. In addition, some studies have shown that the hybrid of PSO and other evolutionary algorithms (EA) also shows promising performance [37]. No matter which collaboration mechanism is used in these types of PSO variants, the main idea is to use different search behaviors of the algorithms involved to improve the search ability, and to share useful information of the algorithms to enhance the development ability.

3. A novel PSO based on hybrid-Learning model

In PSO-HLM, a particle i can learn from E_i, GB, PB_i and Gaussian perturbation based on four learning model (confidence learning model, mild learning model, standard learning model, gaussian learning model). E_i is a potential exemplar particle, it can be selected by the roulette wheel selection from four sub-population exemplar pools (elite pool, potential pool, triple potential pool and fusion pool). During the selection process, crossover probability p_c determines which sub-population exemplar pools to choose for breed E_i. Section 3.1 describes the adaptive change of crossover probability and the random disturbance of inertia weight, Section 3.2 introduces the concept of multi-pool fusion strategy, Section 3.3 details the implementation of hybrid learning model strategy.

3.1. Adaptive crossover probability and disturbance of inertia weight

In the iterative algorithm, fixed parameter values can not fully feed back the information of each stage, resulting in poor performance of the algorithm. In order to adaptively adjust the exploration and exploitation direction according to the current stage information, PSO-HLM adaptively adjust the parameter values according to the stage information.

The crossover probability has a great impact on the diversity of algorithms. When the crossover probability is fixed, the robustness of the algorithm level is poor. Inspired by the characteristics of sine function, the sine function is periodic, it can help to periodically change crossover probability p_c. The possibility of particle dispersion and re-aggregation is greater, and the ability of external search is stronger in the later stagep_c is updated according to Eq (3.1) below.

$$p_c = \alpha_1 - \sin((\text{ CurrentIter / MaxIter }) * \pi) * \alpha_2 \tag{3.1}$$

where p_c is the crossover probability, *CurrentIter* is the current number of iterations, *MaxIter* is the maximum number of iterations, α_1 is set to 0.9, and α_2 is set to 0.4.

Inertia weight w can control the exploration ability of particles, large inertia weight provides sufficient global search, while small inertia weight focuses more on local search. In order to better solve the problem of falling into local optima, the inertia weight is perturbed. This method has the characteristics of large tangent function range and adding random factors, floating in a large range. As a result, particles will have different inertia weights set on each dimension. In Section 3.3, different learning model strategies (confidence model, standard model, mild model and Gaussian model) use different formulas to update inertia weight. Confidence model and Gaussian model uses Eq (3.2) to update the inertia weight of particles, standard model and mild model uses Eq (3.3) to update the inertia weight of particles. Equation (3.2) can help particles focus on small step search, and Eq (3.3) can help particles prefer on big step search. The detail of Eqs (3.2) and (3.3) is as follow:

$$w_d = \epsilon + \eta * \tan((r - 0.5) * \pi) \tag{3.2}$$

$$w_d = \delta + \eta * \tan((r - 0.5) * \pi) \tag{3.3}$$

$$\delta = \frac{2}{\left(\left|2 - (c_1 + c_2) - \sqrt{(c_1 + c_2)^2 - 4 \times (c_1 + c_2)}\right|\right)} \tag{3.4}$$

where d is the dimension, w_d is the inertia weight of the *d-th* dimension. ϵ is a small constriction factor, η is a disturbance coefficient, η=0.005 , $r \in [0,1]$ is a random number. ϵ and δ is a constriction factor, where ϵ is a narrow constriction factor, $\epsilon = 0.45$. δ is a wide constriction factor, according to Clerc's constriction method, c_1 and c_2 are set to 2.05, and the constriction factor δ is approximately 0.7298.

3.2. Multi-pool fusion strategy

PSO is a population-based swarm intelligence algorithm. The diversity of the population can effectively help the algorithm to find the optimal solution quickly. In order to improve the diversity of the population, PSO-HLM divides the population into four sub-populations (elite pool, potential pool, triple potential pool and fusion pool). Different particle pools store particles with different characteristics. For instance, the particle with the best fitness value can only indicate that its current position is better than other particles of the same generation, but it may be close to a local optima and far away from the global optimum. The particle whose fitness value improvement rate is relatively high in two successive generations may be near the local optima or the global optimum.

During the evolutionary process, the elite pool is used to store particles with the best fitness value $f(\mathbf{x})$, the potential pool is used to store particles with the highest improvement rate in a single generation $Ir(\mathbf{x})$, the triple potential pool is used to store particles with the highest average improvement rate in two consecutive generations $Irs(\mathbf{x})$, and the fusion pool is used to store particles with best fitness value and the highest improvement rate at the same time. The improvement rate $Ir(\mathbf{x})$ is defined as:

$$Ir(x_i^t) = \frac{f(x_i^{t-1}) - f(x_i^t)}{e^{\left|x_i^{t-1} - x_i^t\right|}} \tag{3.5}$$

$$Irs\,(x_i^t) = \frac{Ir\left(x_i^{t-1}\right) + Ir\left(x_i^t\right)}{2} \tag{3.6}$$

where $Ir(x_i^t)$ is the improvement rate of particle i at t generation, $f(x_i^{t-1})$ is the fitness value of particle i at t generation, and $\left|x_i^{t-1} - x_i^t\right|$ is the Euclidean distance between x_i^{t-1} and x_i^t. $Irs\left(x_i^t\right)$ is the average improvement rate of particle i in two consecutive generations.

Sub-population pools with different sizes will weaken the fairness of competition and lead to deviation of algorithm advantages. Therefore, the size of each pool is defined as one fourth of the population size, and the size of each pool is equal. They are initialized according to different standards. The elite pool is initialized only by the size of the fitness value and expressed as A_e^t. The potential pool is initialized only by the size of the improvement rate and expressed as A_p^t. The triple potential pool is initialized by the average improvement rate of two consecutive generations, expressed as A_c^t. The fusion pool is initialized by the size of the improvement rate to select the fitness sequence, which is expressed as A_f^t. The four Sub-population pools are represented as follows:

$$A_e^t = \{x_{i1}^t, x_{i2}^t, \ldots, x_{iM}^t \mid f\,(x_{i1}^t) \leq f\,(x_{i2}^t) \leq \ldots \leq f\,(x_{iM}^t)\} \tag{3.7}$$

$$A_p^t = \left\{x_{j1}^t, x_{j2}^t, \ldots, x_{jM}^t \mid Ir\left(x_{j1}^t\right) \leq Ir\left(x_{j2}^t\right) \leq \ldots \leq Ir\left(x_{jM}^t\right)\right\} \tag{3.8}$$

$$A_c^t = \{x_{k1}^t, x_{k2}^t, \ldots, x_{kM}^t \mid Irs\,(x_{k1}^t) \leq Irs\,(x_{k2}^t) \leq \ldots \leq Irs\,(x_{kM}^t)\} \tag{3.9}$$

$$A_f^t = \left\{x_{l1}^t, x_{l2}^t, \ldots, x_{lM}^t \mid \frac{f\left(x_{l1}^t\right) + Ir\left(x_{l1}^t\right)}{2} \leq \frac{f\left(x_{l2}^t\right) + Ir\left(x_{l2}^t\right)}{2} \leq \ldots \leq \frac{f\left(x_{lM}^t\right) + Ir\left(x_{lM}^t\right)}{2}\right\} \tag{3.10}$$

where A_e^t is the elite pool, A_p^t is the potential pool,A_c^t is the triple potential pool, and A_f^t is the fusion pool. t is the number of iterations, M is the size of the particle pool.

In order to take advantage of these four sub-population pools, a potential exemplar E_i is created when updating each dimension of a particle i. In the early stage of the iteration process, four sub-population pools (A_e, A_p, A_c and A_f) is used to generate E_i. In the late stage of the iteration process, three sub-population pools (A_e, A_p and A_f) is used to generate E_i. Because, the particle's fitness value in the triple potential pool does not change much in the late stage of the iteration process, and it does not help to improve the performance of PSO-HLM.

Thus, in the process of iteration, three sub-population pools or four sub-population pools are selected adaptively to generate potential exemplars. In the early stage, the probability of selecting four sub-population pools is high. with the number of iterations increasing, the probability of selecting three sub-population pools is gradually increased. The random number r_f is employed to adjust the sub-population pool selection probability. When $r_f > CurrIter/MaxIter$, Eq (3.12) is used to generate E_i, otherwise, Eq (3.13) is used to generate E_i. In the Eq (3.12), we use roulette wheel selection to select four parents $X_{i_{p1}}$, $X_{i_{p2}}$, $X_{i_{p3}}$ and $X_{i_{p4}}$ from A_e, A_p, A_c and A_f , respectively. The detail of multi-pool fusion operation is shown in Algorithm 1.

$$p_k = \frac{M - k + 1}{\sum_{i=1}^{M} i} \tag{3.11}$$

where p_k is the pre-selection probability of the *k-th* particle, M is the capacity of the excellent parent particle .

$$e_{i,d}^t = \begin{cases} x_{i_{p3},d}^t, & \text{if } r_1 < p_c \text{ and } r_2 < p_c \\ x_{i_{p1},d}^t, & \text{if } r_1 < p_c \text{ and } r_2 > p_c \\ x_{i_{p2},d}^t, & \text{if } r_1 > p_c \text{ and } r_2 < p_c \\ x_{i_{p4},d}^t, & \text{otherwise} \end{cases} \tag{3.12}$$

$$e_{i,d}^t = \begin{cases} x_{i_{p1},d}^t, & \text{if } r_1 < p_c \\ x_{i_{p2},d}^t, & \text{if } r_1 > p_c \text{ and } r_2 < p_c \\ x_{i_{p4},d}^t, & \text{otherwise} \end{cases} \tag{3.13}$$

where $e_{i,d}$ is the position of the *i-th* potential exemplar in the *d-th* dimension, $x_{i_{p1},d}$, $x_{i_{p2},d}$, $x_{i_{p3},d}$ and $x_{i_{p4},d}$ represent the information in d dimension of the *i-th* particle of four sub-population pools, respectively. $p1$, $p2$, $p3$ and $p4$ represents four sub-population pools A_e, A_p, A_c and A_f, respectively. r_1, r_2 is a random number uniformly distributed in [0,1] , p_c is the crossover probability, which is introduced in Section 3.1.

Algorithm 1 The Multi-pool Fusion Operation.

while $i = 1$ to n **do**
  while $j = 1$ to d **do**
    Calculate the probability of selection p_k by Eq (3.11);
  Select $x_{i_{p1}}$ from A_e^t based on the probability p_k;
  Select $x_{i_{p2}}$ from A_e^p based on the probability p_k;
  Select $x_{i_{p3}}$ from A_e^c based on the probability p_k;
  if $r_f > CurrIter/MaxIter$ **then**
    Select $x_{i_{p4}}$ from A_e^f based on the probability p_k;
    Generate potential exemplar $e_{i,j}^t$ according to Eqs (3.1) and (3.12);
  else
    Generate potential exemplar $e_{i,j}^t$ according to Eqs (3.1) and (3.13);

3.3. Hybrid learning model strategy

After applying multi-pool fusion strategy to generate potential exemplar E. The *i-th* particle had three potential exemplars (GB, PB_i and E_i) to learn. In order to increase the local search ability of particles, Gaussian perturbation is also used. Thus, PSO-HLM had four learning models (confidence learning model, standard learning model, mild learning model and Gaussian learning model) at each generation.

In the early stage, If the potential exemplar fitness value $f(e_i^t)$ is less than or equal to global optimal solution $f(gb^t)$, $f(e_i^t) \leq f(gb^t) \leq f(pb_i^t)$. It is considered that E_i is a excellent exemplar and GB may be a local (even a global) optimum solution. Thus, every particles only learning from the potential exemplar, that is the confident learning model, as shown in Eq (3.14). Moreover, GB is replaced by E_i after the particle updates its velocity and position.

If the potential exemplar fitness value $f(e_i^t)$ is less than or equal to the individual optimal solution $f(pb_i^t)$, $f(gb^t) \leq f(e_i^t) \leq f(pb_i^t)$. It is considered that both the global optimal solution gb^t and the potential exemplar had more space to learning, and it can provide more direct guidance information for the *i-th* particle. So, in this case, *i-th* particle not only learn from GB but also learn from E_i, that is mild learning model, as shown in Eq (3.15).

If the potential exemplar fitness value $f(e_i^t)$ is worse than $f(gb^t)$ and $f(gb^t)$, $f(e_i^t) \leq f(pb_i^t) \leq f(gb^t)$. At this time, the particle no longer learns from the potential exemplar, but learns from the two better particles of the global optimal solution GB and the individual optimal solution PB_i, that is standard learning model, as shown in Eq (3.16).

In the middle and late stages of the iteration, $t \geq MaxIter/3$, in order to increase the exploration ability of the population, the Gaussian learning model is applied. It adds Gaussian perturbation while learning from the potential exemplar, so it can effectively jump out of the local optimal. If the fitness value of the perturbed potential exemplar $G(e_i^t)$ is less than or equal to the global optimal solution gb^t, $G(e_i^t) \leq gb^t$. It learning from the excellent particles with a certain step size in a certain direction, which can effectively avoid falling into the local optima prematurely, as shown in Eq (3.19).

According to the aforementioned analysis, the confident, mild, standard and Gaussian learning models detailed as follows are favorable for the exploitation, balanced search, and exploration abilities, respectively.

1) *Confidence Learning Model*: Learning from E_i.

$$v_{i,j}^t = w_d \cdot v_{i,j}^{t-1} + r_{1,j} \cdot \left(e_{i,j}^t - x_{i,j}^{t-1}\right) \tag{3.14}$$

2) *Mild Learning Model*: Learning from E_i and GB.

$$v_{i,j}^t = w_d \cdot v_{i,j}^{t-1} + r_{1,j} \cdot \left(e_{i,j}^t - x_{i,j}^{t-1}\right) + r_{2,j} \cdot \left(gb_j^t - x_{i,j}^{t-1}\right) \tag{3.15}$$

3) *Standard Learning Model*: Learning from PB_i and GB.

$$v_{i,j}^t = w_d \cdot v_{i,j}^{t-1} + r_{1,j} \cdot \left(pb_{i,j}^t - x_{i,j}^{t-1}\right) + r_{2,j} \cdot \left(gb_j^t - x_{i,j}^{t-1}\right) \tag{3.16}$$

4) *Gaussian Learning Model*: Learning from E_i and Gaussian perturbation.

$$Gas_i = r_1 * \mathcal{N}\left(\mu, \sigma^2\right) \tag{3.17}$$

$$G\left(e_i^t\right) = |\mathrm{Gas}_i| * f\left(e_i^t\right) \tag{3.18}$$

$$v_{i,j}^t = w_d \cdot v_{i,j}^{t-1} + r_{1,j} \cdot \left(e_{i,j}^t - x_{i,j}^{t-1}\right) + 0.001 \cdot Gas_i \tag{3.19}$$

where $v_{i,j}^t$ is the velocity of the *j-th* dimension of the *i-th* particle at t generation. w_d is the inertia weight by Eqs (3.2) and (3.3). $r_{1,j}$, $r_{2,j}$ are random number uniformly distributed in [0,1], $x_{i,j}^{t-1}$ is the position of the *j-th* dimension of the *i-th* particle at $t-1$ generation. pb is the individual optimal solution, gb is the global optimal solution, and e is the potential exemplar. Gas_i is the Gaussian random number of the *i-th* particle, $\mathcal{N}$ is a standard Gaussian distribution, μ is the mean value and σ is the variance. $G(e_i^t)$ is a Gaussian perturbation value of the *i-th* potential exemplar at t generation. $f(e_i^t)$ is the fitness value of the *i-th* potential exemplar at t generation.

In the last, the potential exemplar will be reused, and it is more likely to become key role models for other particles to learn. The reuse formula is Eq (3.20).

$$pb_o^t = \begin{cases} e_i^t, & \text{if } G(e_i^t) \leq G(e_j^t) \\ e_j^t, & \text{otherwise} \end{cases} \tag{3.20}$$

where pb_o is the individually optimal solution for the *o-th* particles, o is the particle random number, e_i is the position of the *i-th* offspring, $G(e_i)$ and $G(e_j)$ is the value of the *i-th* and *j-th* potential exemplar fitness value after Gaussian perturbation. t is the number of iterations.

3.4. Framework of PSO-HLM

By incorporating the aforementioned components, The details of the pseudocode of PSO-HLM are shown in Algorithm 2. There are several steps. First, the parameters (population size, the number of iterations and the mutation rate) are initialized. Then the position information of each particle is initialized and the fitness value of each particle is calculated. The particles with good fitness are selected to initialize the global optimal solution and the individual optimal solution. In iteration loop, different individuals from different particle pools are used for crossover-fusion to generate potential exemplar particles, as shown by Algorithm 1. According to the fitness of potential exemplar particles, the dominant particles are more likely to be selected, and the velocity and position of particles is updated through the corresponding learning model. In the early stage, the learning model is selected according to the fitness value of potential exemplar particles. After a specific stage, the fitness value of potential exemplar particles is perturbed by the Gaussian distribution. An appropriate learning model is selected to update the velocity and position of potential exemplar particles, according to the disturbed fitness value and the original fitness value. Then, the global optimal solution is replaced by the generated optimal potential exemplar particles. Through the transformation and stage adaptation of different learning models, the best solutions can be found.

3.5. Complexity analysis

We denote the population size is N, the dimensionality of the problem is D. First, the relevant parameters of the algorithm, the velocity and position of particle are initialized. In the initialization process, the time complexity of PSO-HLM related parameters is $O(K)$ and K is constant. The time complexity of initialization velocity and position is $O(N \times D)$. It is concluded that the time complexity of initialization is $O(N \times D)$. Then, the excellent individuals are crossover fused to generate potential exemplar. Moreover, the information of each dimension of potential exemplar needs to be uniquely selected, so the time complexity of crossover is $O(N^2 \times D)$. After that, particles need to choose the corresponding learning model according to velocity and position. Since each particle selects the learning model only once, the time complexity of the selection operation is $O(N)$. Because each operation of the algorithm is completed in parallel, the total time complexity of each is $O(K + N \times D + N^2 \times D + N)$. Therefore, it is preliminarily determined that the time complexity of PSO-HLM is $O(N^2 \times D)$. If the number of iterations of PSO-HLM is K, the asymptotic upper bound of the algorithm time complexity is $O(K \times N^2 + K \times D)$. As a result, the overall time complexity of PSO-HLM is $O(N^2 \times D)$.

Algorithm 2 The Fusion Operation.

t=1;
while $i = 1$ to n **do**
 Initialization particle v_i^t and x_i^t;
 Refresh pbest, $pb_i^t = x_i^t$;
Update gb, $gb^t = pb_i^t$;
while $CurrentIter \leq MaxIter$ **do**
 $t = t + 1$;
 Calculation improvement rate $Ir(\mathbf{X}_i^t)$, $Irs(\mathbf{X}_i^t)$ by Eqs (3.5) and (3.6);
 Update parent sequence p_k, A_e^t, A_p^t, A_c^t and A_f^t by Eqs (3.7)–(3.11);
 while $i = 1$ to N **do**
  Generation of potential exemplar e_i^t by Algorithm 1;
  if $CurrentIter/MaxIter > \frac{1}{3}$ **then**
   Perturbation of potential exemplar fitness value $G(e_i^t)$ by Eqs (3.17) and (3.18);
  if $G(e_i^t) \leq f(gb^t)$ **then**
   Update w_d and v_i^t based on Eqs (3.2) and (3.19);
  else
   if $f(e_i^t) \leq f(gb^t)$ **then**
    Update w_d and v_i^t based on Eqs (3.2) and (3.14);
    Update gb^t;
   else if $f(e_i^t) \leq f(pb_i^t)$ **then**
    Update w_d and v_i^t based on Eqs (3.3) and (3.15);
   else
    Update w_d and v_i^t based on Eqs (3.3) and (3.16);
  Update position $x_i^t = x_i^{t-1} + v_i^t$
 Update gb^t and location information;
 The reuse of the offspring e_i^t is based on Eq (3.20);

Table 1. The information of the benchmark functions.

No.	Function	Search space
F1	Sphere	$[-100, 100]^D$
F2	Schwefel P2.22	$[-10, 10]^D$
F3	Schwefel P1.2	$[-100, 100]^D$
F4	Schwefel P2.6 with Global optimum on Bounds	$[-100, 100]^D$
F5	Shifted Sphere	$[-100, 100]^D$
F6	Shifted Schwefel P1.2	$[-100, 100]^D$
F7	Shifted Schwefel P1.2 with Noise in fitness	$[-100, 100]^D$
F8	Shifted Rotated High Conditioned Elliptic	$[-100, 100]^D$
F9	Ackley	$[-32, 32]^D$
F10	Schwefel	$[-500, 500]^D$
F11	Rastrigin	$[-5.12, 5.12]^D$
F12	Noncont Rastrigin	$[-5.12, 5.12]^D$
F13	Weierstrass	$[-0.5, 0.5]^D$
F14	Penalized	$[-50, 50]^D$
F15	Salomon	$[-100, 100]^D$
F16	Pathological	$[-100, 100]^D$
F17	Rosenbrock	$[-30, 30]^D$
F18	Griewank	$[-600, 600]^D$
F19	Expanded Extended Griewankplus Rosenbrock	$[-3, 1]^D$
F20	Schwefel P2.13	$[-\pi, \pi]^D$
F21	Shifted Rastrigin	$[-5.12, 5.12]^D$
F22	Shifted Noncont Rastrigin	$[-5.12, 5.12]^D$
F23	Shifted Rosenbrock	$[-100, 100]^D$
F24	Shifted Rotated Expanded Scaffer F6	$[-100, 100]^D$
F25	Shifted Rotated Griewank without Bounds	$[-600, 600]^D$
F26	Shifted Rotated Ackley with Global optimum on bounds	$[-32, 32]^D$
F27	Shifted Rotated Rastrigin	$[-5.12, 5.12]^D$
F28	Shifted Rotated Noncont Rastrigin	$[-5.12, 5.12]^D$
F29	Shifted Rotated Weierstrass	$[-0.5, 0.5]^D$
F30	Shifted Rotated Salomon	$[-100, 100]^D$

4. Experimental studies

4.1. Benchmark functions and peer algorithms

In order to fully evaluate the performance of PSO-HLM and ensure the fairness of the experiment, the same benchmark function suite (30 functions) is used, followed by TAPSO [18]. This benchmark function suite contains 8 unimodal functions and 22 multimodal functions. The unimodal functions have the basic unimodal functions (F1–F4) and the modified unimodal functions (F5–F8). The multimodal functions have the basic multimodal functions (F9–F20) and modified multimodal functions (F21–F30), the detailed information was shown in Table 1. The test results were compared and analyzed with 5 state-of-the-art algorithms (CLPSO [22], FDR-PSO [32], PPSO [38], TAPSO [18] and XPSO [35]). All algorithms were run on MATLAB 2019b and the processor was an Intel(R) Core (TM) i7-9750H CPU @ 2.60 GHz 2.59 GHz. In addition, a number of experiments were conducted aimed at analyzing and illustrating the effectiveness of the newly introduced strategies, as detailed in Section 4.5.

4.2. Experimental parameter setting

The population size N of PSO-HLM is 60 and the p_m is 0.02. To be fair, for all algorithms, the parameters uses the default settings in its original paper, the detail information of experimental parameter settings is shown in Table 2. All algorithms are given a random initial position and velocity. The fitness evaluation sizes was set to $10000 * D$. For each algorithm, each benchmark test function was run 30 times, and the mean and standard deviation (std) of each algorithm are used for comparison.

Each algorithm is independently performed for 30 times for the propose of statistical comparisons, and for each run, the mean value (Mean) and standard variance (Std) of the solution is recorded, the best results of each function are presented in bold. In the comparative experiments, the best result is bolded. Results of PSO-HLM are compared with those of CLPSO, FDR-PSO, PPSO, TAPSO and XPSO, respectively, by Wilcoxon rank sum test at the significance level of 0.05. The marker "-" is worse than the results of PSO-HLM, "+" is better than the results of PSO-HLM and "≈" is equivalent to the results of PSO-HLM.

Table 2. Parameter setting of each algorithm.

Algorithm	Population size	Experimental parameter settings
CLPSO	N = 60	$w_{max} = 0.9, w_{min} = 0.4, c1 = 1.49445, c2 = 0$
FDR-PSO	N = 60	$w = [0.4, 0.9], c_1 = 1, c_2 = 1, c_3 = 2$
PPSO	N = 40	$w = 0, c_1 = \left\lvert\cos\theta_i^{Iter}\right\rvert^{2*\sin\theta_i^{Iter}}, c_2 = \left\lvert\sin\theta_i^{\text{CurrentIter}}\right\rvert^{2*\cos\theta_i^{\text{CurrentIter}}}$
TAPSO	N = 60	$w = 0.7298, p_c = 0.5, p_m = 0.02, M = N/4$
XPSO	N = 60	$w = [0.4, 0.9], c_1 = 1, c_2 = 0.5, c_3 = 0.5$
PSO-HLM	N = 60	w = Eqs (3.2) or (3.3), p_c = Eq (3.1), $p_m = 0.02$

4.3. Statistical results of solutions

4.3.1. Analysis of experimental results on 30-D

In this experiment, five state-of-the-art algorithms (CLPSO [22], FDR-PSO [32], PPSO [38], TAPSO [18] and XPSO [35]), are used for comparisons with PSO-HLM on the 30-D problems. The results was reported in Table 3.

1) Unimodal Functions (F1–F8): from Table 3, it can be seen that PSO-HLM has the best mean value in many functions. On the F4 and F7 functions, the results obtained by PSO-HLM are always the most promising for most functions, so that the introduction of different improvement rates increases the likelihood of finding the best and the variety of learning models allows for an adequate search of the problem space. However, on the F4 and F7 functions, XPSO has good mean values, which shows

that the forgetting ability of the particles can well help these two functions need to find the optimum.

Table 3. The results of the algorithm on 30-D.

No.		CLPSO	FDR-PSO	PPSO	TAPSO	XPSO	PSO-HLM
F1	mean	1.56E-36 +	3.21E-124 +	2.01E-14 +	1.94E-160 +	4.53E-132 +	**3.62E-180**
	std	1.45E-36 +	1.76E-123 +	2.27E-14 +	6.24E-160 +	2.48E-131 +	**0.00E+00**
F2	mean	7.76E-24 +	2.48E-57 +	1.30E-04 +	1.34E-78 +	2.77E-41 +	**3.81E-91**
	std	4.68E-24 +	1.25E-56 +	1.86E-04 +	4.54E-78 +	1.52E-40 +	**1.39E-90**
F3	mean	1.42E+02 +	2.28E-10 +	8.27E-08 +	2.21E-26 +	1.62E-14 +	**1.17E-31**
	std	8.41E+01 +	3.71E-10 +	1.38E-07 +	7.95E-26 +	5.25E-14 +	**2.97E-31**
F4	mean	4.00E+03 +	3.09E+03 +	6.39E+03 +	3.91E+03 +	**3.02E+03** +	3.41E+03
	std	4.41E+02 +	5.10E+02 +	2.20E+03 +	1.15E+03 +	**7.88E+02** +	9.02E+02
F5	mean	**0.00E+00** ≈	**0.00E+00** ≈	2.53E+01 +	**0.00E+00** ≈	3.53E-29 +	**0.00E+00**
	std	**0.00E+00** ≈	**0.00E+00** ≈	1.17E+02 +	**0.00E+00** ≈	8.28E-29 +	**0.00E+00**
F6	mean	1.09E+03 +	2.57E-11 +	6.36E+01 +	6.46E-20 +	8.25E-06 +	**6.59E-27**
	std	2.01E+02 +	7.86E-11 +	1.61E+02 +	3.50E-19 +	1.75E-05 +	**4.96E-27**
F7	mean	7.52E+03 +	3.41E+02 +	2.23E+03 +	8.69E+03 +	**9.70E+01** +	1.13E+03
	std	1.65E+03 +	2.02E+02 +	1.22E+03 +	4.07E+03 +	**6.41E+01** +	8.29E+02
F8	mean	1.74E+07 +	4.88E+05 +	1.61E+06 +	2.23E+05 +	3.61E+06 +	**1.67E+05**
	std	4.30E+06 +	2.26E+05 +	3.63E+06 +	1.17E+05 +	4.17E+06 +	**7.43E+04**
F9	mean	7.82E-15 +	1.97E-14 +	2.69E-08 +	**5.57E-15** +	7.34E-15 +	6.63E-15
	std	1.45E-15 +	4.82E-15 +	1.26E-08 +	**1.79E-15** +	1.30E-15 +	1.23E-15
F10	mean	7.90E+00 +	3.24E+03 +	8.31E+02 +	1.82E-13 +	3.19E+03 +	**0.00E+00**
	std	3.00E+01 +	4.79E+02 +	7.54E+02 +	5.55E-13 +	4.61E+02 +	**0.00E+00**
F11	mean	3.32E-02 +	2.57E+01 +	1.98E-13 +	**0.00E+00** ≈	2.47E+01 +	**0.00E+00**
	std	1.82E-01 +	8.22E+00 +	1.41E-13 +	**0.00E+00** ≈	8.58E+00 +	**0.00E+00**
F12	mean	**0.00E+00** ≈	1.04E+01 +	1.54E-13 +	**0.00E+00** ≈	4.80E+00 +	**0.00E+00**
	std	**0.00E+00** ≈	4.70E+00 +	7.50E-14 +	**0.00E+00** ≈	4.40E+00 +	**0.00E+00**
F13	mean	**0.00E+00** +	2.00E-03 +	5.29E-02 +	2.83E-02 +	7.70E-03 +	8.94E-04
	std	**0.00E+00** +	1.05E-02 +	4.17E-02 +	4.31E-02 +	1.49E-02 +	1.50E-03
F14	mean	**1.57E-32** +	**1.57E-32** +	3.60E-13 +	1.59E-32 ≈	1.93E-32 +	1.59E-32
	std	**1.11E-47** +	**1.11E-47** +	3.25E-13 +	9.43E-34 ≈	1.63E-32 +	9.43E-34
F15	mean	**2.40E-01** +	2.90E-01 +	1.38E+00 +	5.83E-01 +	2.90E-01 +	4.80E-01
	std	**4.78E-02** +	6.62E-02 +	5.89E-01 +	1.39E-01 +	4.81E-02 +	7.61E-02
F16	mean	**8.80E-01** +	5.79E+00 +	5.56E+00 +	3.38E+00 +	8.52E+00 +	3.64E+00
	std	**3.91E-01** +	9.09E-01 +	3.70E+00 +	7.82E-01 +	1.34E+00 +	7.33E-01
F17	mean	2.13E+00 +	3.89E+00 +	2.17E+01 +	5.22E-12 +	6.14E+00 +	**3.53E-12**
	std	5.61E+00 +	3.09E+00 +	5.19E-01 +	8.08E-12 +	1.40E+01 +	**5.04E-12**
F18	mean	**0.00E+00** +	1.25E-02 +	2.47E-05 +	5.90E-03 +	9.20E-03 +	5.10E-03
	std	**0.00E+00** +	1.73E-02 +	1.35E-04 +	8.30E-03 +	7.40E-03 +	1.02E-02
F19	mean	1.65E+00 +	2.90E+00 +	9.61E+00 +	1.14E+00 +	2.85E+00 +	**1.12E+00**
	std	1.86E-01 +	8.13E-01 +	3.37E+00 +	2.47E-01 +	7.94E-01 +	**1.78E-01**
F20	mean	1.81E+04 +	6.95E+03 +	4.38E+04 +	2.17E+03 +	8.37E+03 +	**9.95E+02**
	std	5.19E+03 +	8.42E+03 +	5.25E+04 +	4.32E+03 +	7.36E+03 +	**1.40E+03**
F21	mean	1.33E-01 +	2.85E+01 +	6.62E+01 +	**0.00E+00** ≈	2.69E+01+	**0.00E+00**
	std	3.44E-01 +	9.05E+00 +	2.39E+01 +	**0.00E+00** ≈	8.47E+00 +	**0.00E+00**
F22	mean	**0.00E+00** ≈	1.11E+01 +	2.30E+01 +	**0.00E+00** ≈	5.80E+00 +	**0.00E+00**
	std	**0.00E+00** ≈	6.04E+00 +	1.61E+01 +	**0.00E+00** ≈	4.59E+00 +	**0.00E+00**
F23	mean	9.89E+00 +	1.53E+01 +	2.00E+02 +	5.47E-04 +	2.89E+01 +	**4.28E-10**
	std	1.76E+01 +	3.30E+01 +	3.08E+02 +	2.40E-03 +	5.24E+01 +	**1.45E-09**
F24	mean	1.29E+01 +	1.18E+01 +	1.32E+01 +	1.20E+01 +	**1.17E+01** +	1.21E+01
	std	2.30E-01 +	5.17E-01 +	3.58E-01 +	5.90E-01 +	**5.54E-01** +	6.25E-01
F25	mean	1.33E-01 +	**1.60E-02** +	3.69E+00 +	1.82E-02 +	1.19E+00 +	2.53E-02
	std	4.58E-02 +	**1.67E-02** +	1.43E+01 +	1.83E-02 +	4.34E-01 +	3.31E-02
F26	mean	2.09E+01 +	2.09E+01 +	**2.02E+01** +	2.03E+01 +	2.09E+01 +	2.04E+01
	std	5.49E-02 +	6.44E-02 +	**2.08E-01** +	4.01E-01 +	5.64E-02 +	4.65E-01
F27	mean	1.24E+02 +	5.91E+01 +	3.71E+02 +	7.34E+01 +	5.57E+01 +	**5.46E+01**
	std	1.63E+01 +	2.07E+01 +	8.45E+01 +	1.88E+01 +	3.96E+01 +	**1.51E+01**
F28	mean	1.32E+02 +	8.03E+01 +	3.74E+02 +	9.00E+01 +	9.14E+01 +	**7.10E+01**
	std	2.09E+01 +	2.14E+01 +	9.80E+01 +	2.81E+01 +	3.45E+01 +	**1.71E+01**
F29	mean	2.81E+01 +	1.84E+01 +	3.52E+01 +	1.99E+01 +	**1.37E+01** +	1.70E+01
	std	1.91E+00 +	3.72E+00 +	3.55E+00 +	3.56E+00 +	**3.66E+00** +	2.49E+00
F30	mean	**2.40E-01** +	2.97E-01 +	1.43E+00 +	5.43E-01 +	2.77E-01 +	4.80E-01
	std	**4.96E-02** +	7.65E-02 +	4.86E-01 +	1.36E-01 +	5.68E-02 +	7.13E-02
	- / ≈ / +	21/3/6	22/1/7	28/0/2	20/5/5	24/0/6	

2) Basic Multimodal Functions (F9–F20): as shown in Table 3, the ones with the most optimal means are PSO-HLM and CLPSO. On F9, TAPSO has the best mean value, its performance is better than PSO-HLM. Because the division condition of PSO-HLM's diverse learning model is not the most suitable, thus it leads to the omission of the optimal solution. However, on F10, PSO-HLM has the best optimal value, indicating that the diverse learning models and the multi-pool fusion can improve performance effectively. On F14, F15, F16 and F18, CLPSO has the best average mean value, it is shown that uninterrupted learning toward optimal particles interferes with the search for optimal solutions, and that integrated reviews and different choices are more conducive to the solution of such problems. The improvement rate is also very important as can be seen from the F17 and F19 functions. On F20, PSO-HLM significantly outperforms than other algorithm. In general, it is found that PSO-HLM is the most promising algorithm compared to other algorithms.

3) Modified Multimodal Function (F21–F30): it can be seen from the data in Table 3 that TAPSO can find the optimal solution only on F21 and F22. PSO-HLM not only finds the optimal solution on F21 and F22, but also outperforms the other five state-of-the-art algorithms on three functions F23, F27 and F28. It shown that for multimodal problems, more diversified learning modes and diversified particle models can effectively alleviate premature local concentration of particle swarm. On F22 and F30, the results of CLPSO illustrate the importance of learning model selection. The results of mainstream algorithms further show that the performance of the algorithm can be improved by appropriate learning models. In short, an appropriate learning model and an appropriate conversion time are the key to the superiority of the algorithm.

In order to better evaluate the performance of all algorithms, the Friedman test was conducted individually for all algorithms in this paper. The mean value of the 30 test functions of all algorithms were run in the Friedman test, the average ranking value smaller indicates better performance. From Table 7, it is evident that six algorithms at 30-D can be sorted into the following order: PSO-HLM, TAPSO, CLPSO, FDR-PSO, XPSO and PPSO. PSO-HLM has the best average ranking, PPSO has the worst results.

Table 4. Average rankings achieved by Friedman test at 30-D.

No.	Algorithms	The average Ranking
1	PSO-HLM	2.03
2	TAPSO	2.77
3	CLPSO	3.60
4	FDR-PSO	3.62
5	XPSO	3.78
6	PPSO	5.20

4.3.2. Convergence speed analysis on 30-D

It is necessary to measure the speed of obtaining the global optimal solution as well as the accuracy of the solution. In this section, convergence analysis and speed comparison are performed on the experimental data of Section 4.3.1. The convergence curves plots of all algorithms are drawn for each function on the 30-D functions, and it is shown in Figures 2–6.

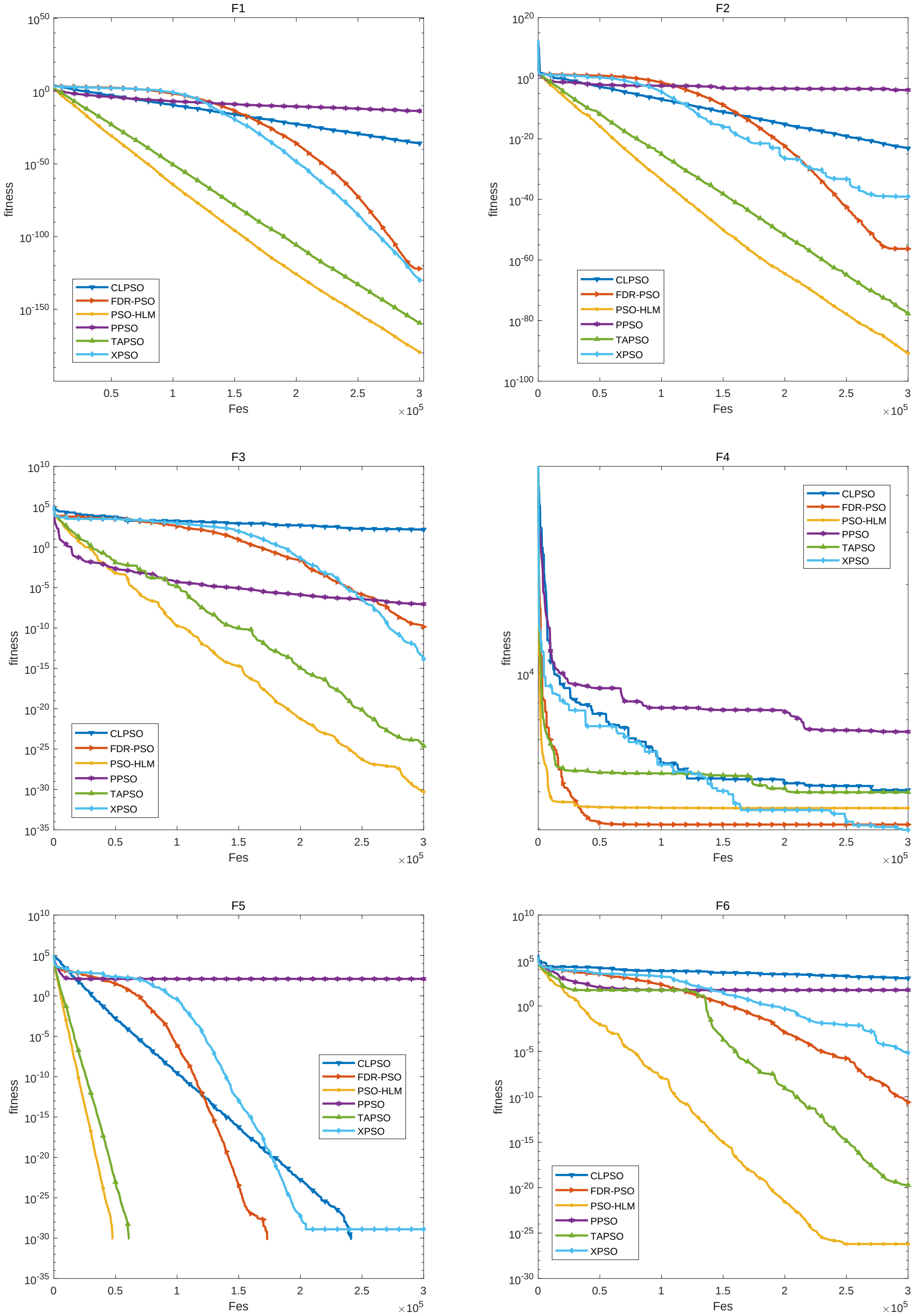

Figure 2. The convergence curves on F1–F15.

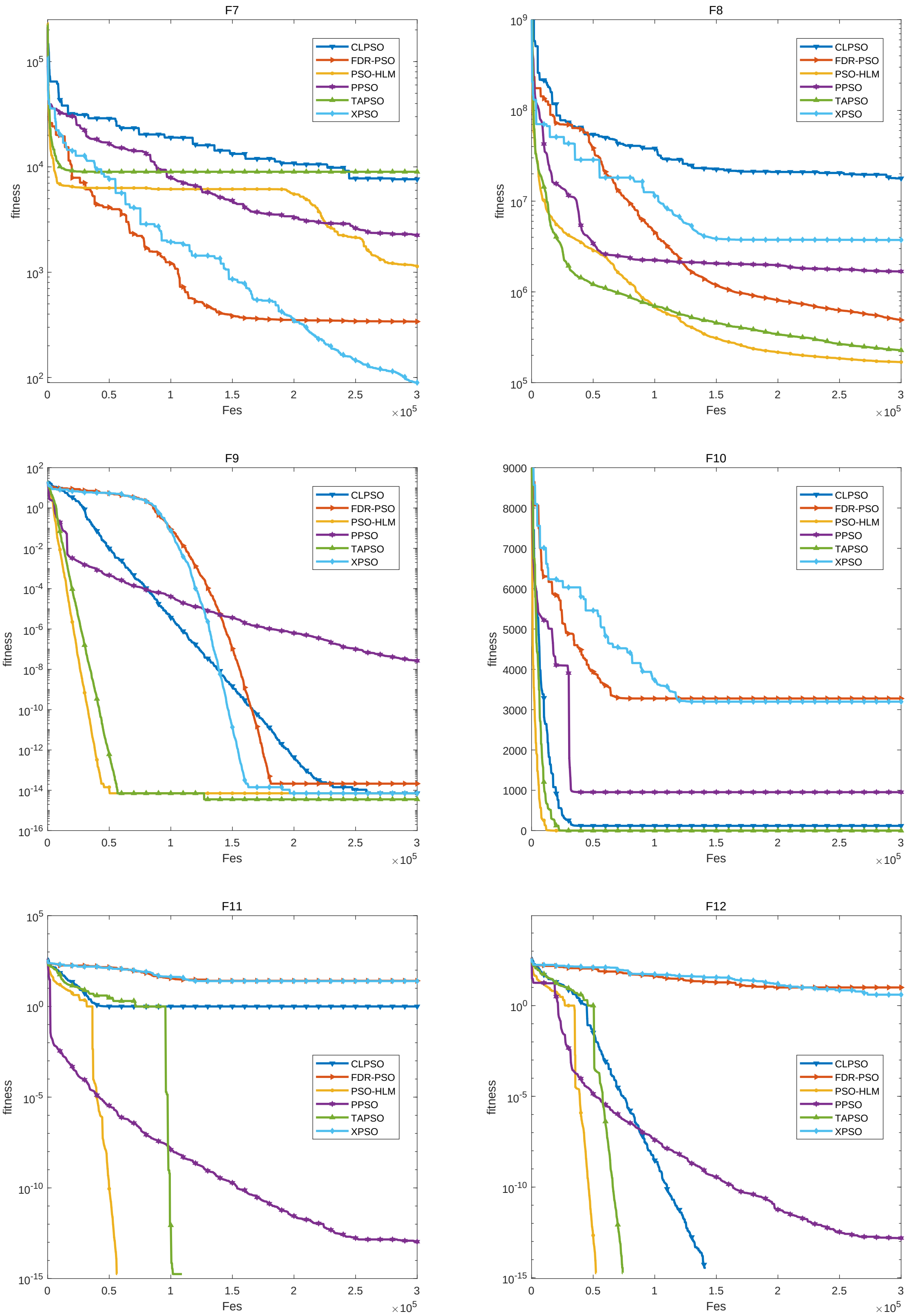

Figure 3. The convergence curves on F1–F15.

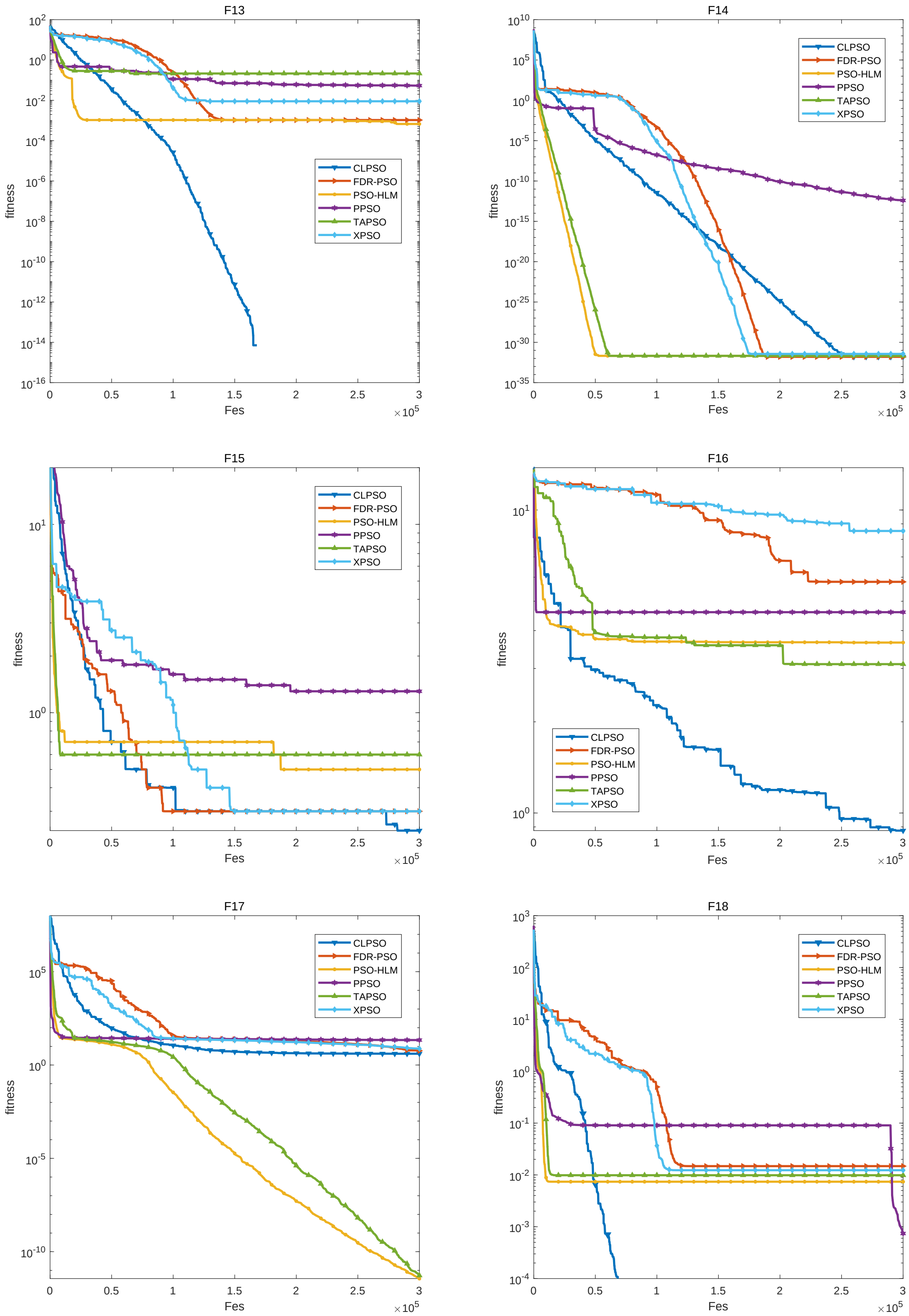

Figure 4. The convergence curves on F1–F15.

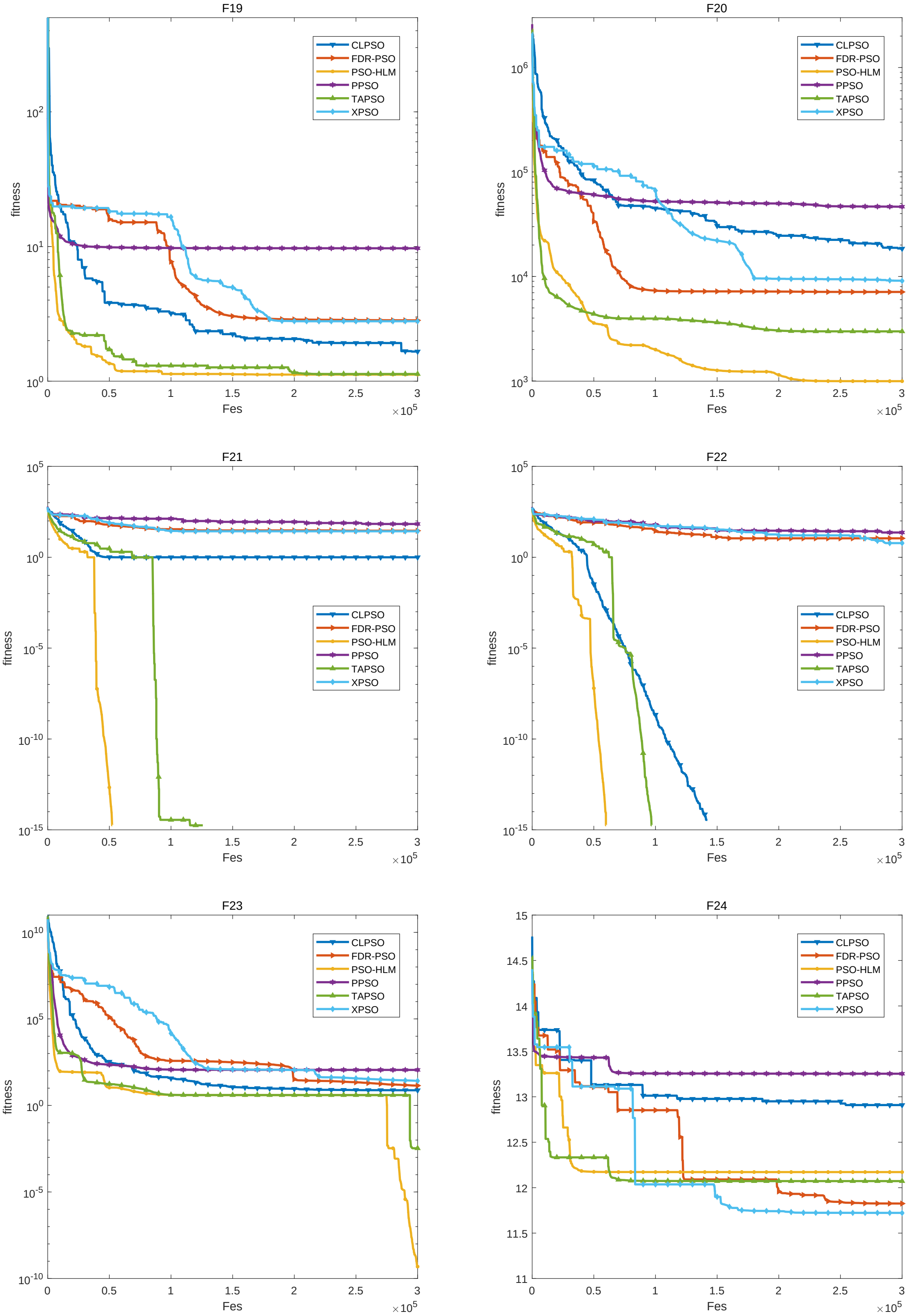

Figure 5. The convergence curves on F16–F30.

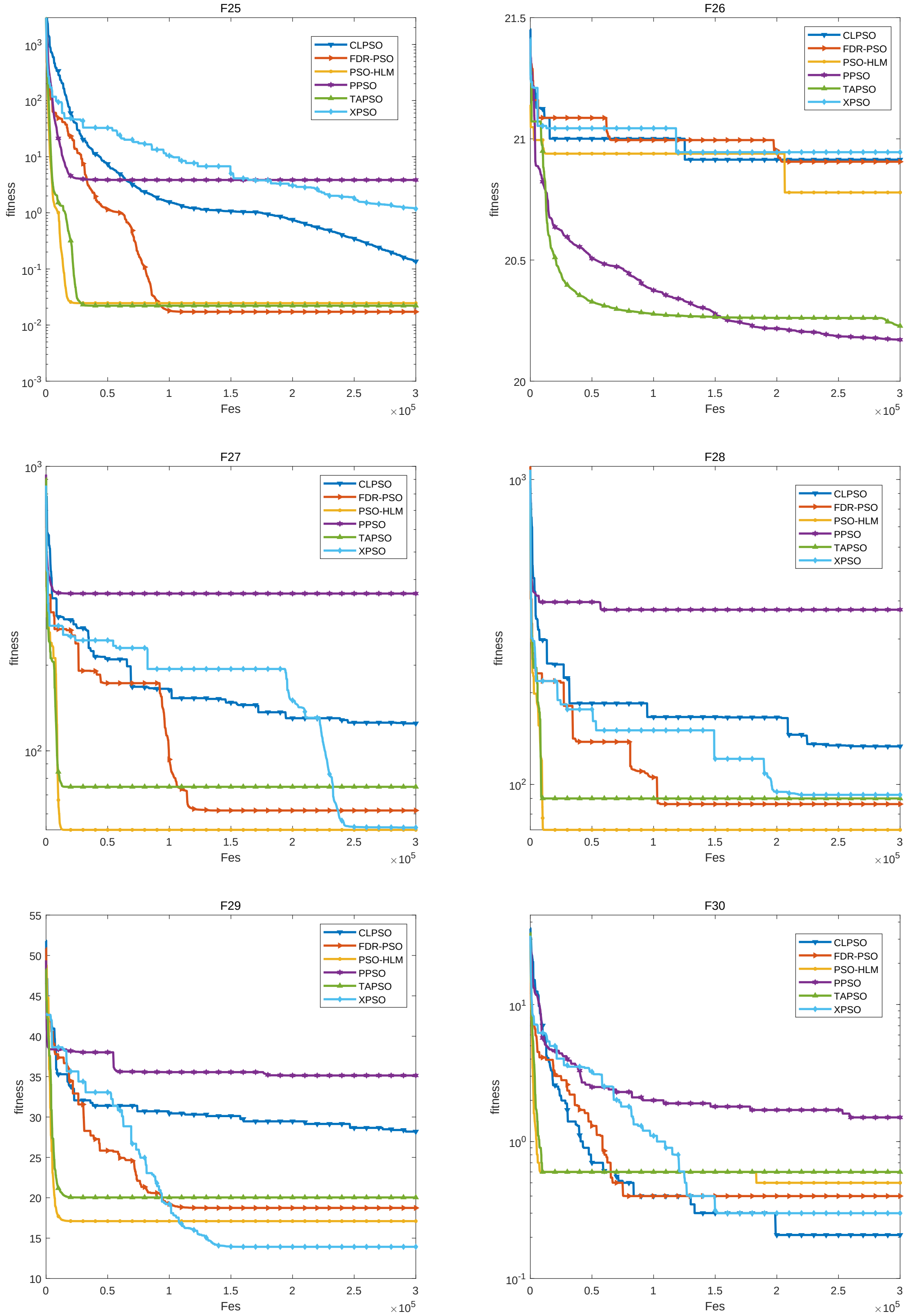

Figure 6. The convergence curves on F16–F30.

Table 5. The results of the algorithm on 50-D.

No.		CLPSO	FDR-PSO	PPSO	TAPSO	XPSO	PSO-HLM
F1	mean	6.68E-26 +	5.16E-101 +	1.95E-13 +	**9.31E-191**-	1.81E-130 +	1.87E-144
	std	1.55E-26 +	5.84E-101 +	1.61E-13 +	**0.00E+00** -	3.13E-130 +	3.24E-144
F2	mean	4.51E-17 +	1.83E-44 +	1.66E-04 +	3.60E-45 +	4.99E-32 +	**9.37E-49**
	std	5.04E-18 +	2.38E-44 +	1.21E-04 +	6.22E-45 +	8.15E-32 +	**9.05E-50**
F3	mean	4.34E+02 +	1.70E-03 +	2.49E-07 +	4.50E-08 +	1.51E-08 +	**3.58E-13**
	std	5.18E+02 +	5.34E-04 +	1.73E-07 +	7.15E-08 +	1.22E-08 +	**2.53E-13**
F4	mean	9.56E+03 +	7.00E+03 -	1.57E+04 +	9.41E+03 +	**6.84E+03** -	7.04E+03
	std	6.20E+02 +	5.65E+02 -	2.12E+03 +	2.52E+03 +	**1.36E+03** -	7.03E+02
F5	mean	**0.00E+00**≈	**0.00E+00**≈	1.31E-11 +	2.10E-29 +	5.05E-28 +	**0.00E+00**
	std	**0.00E+00**≈	**0.00E+00**≈	6.53E-12 +	2.63E-29 +	3.53E-28 +	**0.00E+00**
F6	mean	6.55E+03 +	3.93E-04 +	2.18E+02 +	5.65E-06 +	1.58E-01 +	**2.77E-12**
	std	1.52E+03 +	4.71E-04 +	1.96E+02 +	7.26E-06 +	1.89E-01 +	**7.60E-13**
F7	mean	2.68E+04 +	1.15E+04 -	1.93E+04 +	2.91E+04 +	**2.15E+03** -	1.46E+04
	std	3.04E+03 +	4.69E+03 -	5.46E+03 +	5.15E+03 +	**5.66E+02** -	1.78E+03
F8	mean	5.19E+07 +	6.29E+05 +	6.01E+06 +	9.36E+05 +	4.43E+07 +	**1.56E+05**
	std	1.14E+06 +	2.25E+05 +	4.71E+06 +	9.16E+05 +	1.79E+07 +	**5.18E+04**
F9	mean	8.05E-14 +	4.62E-14 +	6.80E-08 +	1.42E-14 +	1.66E-14 +	**9.47E-15**
	std	1.35E-14 +	1.63E-14 +	1.77E-08 +	1.93E-30 +	4.10E-15 +	**4.10E-15**
F10	mean	**1.82E-11**-	5.42E+03 +	3.23E+03 +	2.30E-11 +	6.67E+03 +	2.06E-11
	std	**0.00E+00** -	1.03E+03 +	4.15E+03 +	8.40E-12 +	3.77E+02 +	4.20E-12
F11	mean	**0.00E+00** ≈	4.41E+01 +	3.90E-13 +	1.78E-15 +	6.20E+01 +	**0.00E+00**
	std	**0.00E+00** ≈	1.70E+01 +	8.62E-14 +	1.78E-15 +	1.39E+01 +	**0.00E+00**
F12	mean	**0.00E+00**≈	2.40E+01+	5.07E-13+	**0.00E+00**≈	6.67E+00 +	**0.00E+00**
	std	**0.00E+00**≈	6.56E+00+	6.45E-14+	**0.00E+00**≈	3.51E+00 +	**0.00E+00**
F13	mean	**0.00E+00**-	1.64E+00 +	5.87E-02 -	1.28E-01 -	1.90E-01 +	1.28E-01
	std	**0.00E+00** -	1.57E+00+	8.39E-02 -	3.87E-02 -	2.10E-01 +	6.91E-02
F14	mean	3.96E-28+	**9.42E-33**≈	1.74E-13+	1.25E-32+	**9.42E-33**≈	**9.42E-33**
	std	8.21E-29+	**0.00E+00**≈	6.68E-14+	3.10E-33+	**0.00E+00**≈	**0.00E+00**
F15	mean	**3.68E-01** -	4.67E-01 -	2.57E+00+	1.60E+00 +	4.33E-01 -	6.33E-01
	std	**5.48E-02**-	1.53E-01 -	1.53E-01 +	2.65E-01 +	5.77E-02 -	5.77E-02
F16	mean	**3.18E+00**-	1.01E+01 +	4.84E+00 -	6.88E+00 +	1.91E+01 +	5.83E+00
	std	**4.95E-01**-	5.45E-01 +	8.37E+00 -	1.05E+00 +	8.28E-01 +	1.03E+00
F17	mean	2.71E+00 +	2.51E+01 +	4.17E+01 +	4.82E-14 +	4.12E+01 +	**1.78E-15**
	std	2.57E+00 +	4.33E+01 +	3.01E-01 +	8.25E-14 +	4.08E+01 +	**1.59E-15**
F18	mean	**0.00E+00** -	7.40E-03-	1.37E-10-	3.07E-02-	3.30E-03-	4.40E-02
	std	**0.00E+00**-	1.28E-02-	7.59E-11-	5.32E-02-	5.70E-03-	5.27E-02
F19	mean	3.05E+00+	5.85E+00+	1.92E+01+	1.98E+00+	5.63E+00+	**1.53E+00**
	std	2.27E-01+	1.20E+00+	6.11E+00+	1.52E-01+	4.55E-01+	**9.56E-02**
F20	mean	5.53E+04+	4.69E+04+	4.18E+05+	1.16E+04+	5.55E+04+	**1.05E+03**
	std	1.20E+04+	3.00E+04+	4.23E+05+	1.51E+04+	2.08E+04+	**2.96E+02**
F21	mean	3.32E-01+	7.79E+01+	1.85E+02+	2.37E-15+	6.40E+01+	**0.00E+00**
	std	5.74E-01+	3.77E+00+	1.01E+01+	2.05E-15+	1.15E+00+	**0.00E+00**
F22	mean	**0.00E+00**≈	3.23E+01+	4.80E+01+	**0.00E+00**≈	8.67E+00+	**0.00E+00**
	std	**0.00E+00**≈	1.30E+01+	3.94E+01+	**0.00E+00**≈	8.96E+00+	**0.00E+00**
F23	mean	2.13E+01+	3.22E+01+	8.56E+01+	6.96E-05+	1.49E+01+	**4.65E-11**
	std	3.61E+01+	3.91E+01+	8.21E+01+	1.21E-04+	1.39E+01+	**7.64E-11**
F24	mean	2.28E+01+	2.15E+01+	2.30E+01+	2.12E+01+	2.18E+01+	**2.11E+01**
	std	1.36E-01+	3.35E-01+	5.57E-01+	1.18E+00+	5.61E-01+	**6.75E-01**
F25	mean	1.09E-01+	4.10E-03+	9.40E-01+	2.30E-02+	2.70E+00+	**2.50E-03**
	std	1.24E-02+	7.10E-03+	1.61E+00+	2.48E-02+	2.77E+00+	**4.30E-03**
F26	mean	2.11E+01+	2.11E+01+	2.02E+01+	2.04E+01+	2.11E+01+	**2.00E+01**
	std	3.70E-03+	4.43E-02+	7.85E-02+	5.98E-01+	3.32E-02+	**9.90E-03**
F27	mean	2.94E+02+	1.15E+02+	8.77E+02+	1.48E+02+	6.00E+01+	**1.04E+02**
	std	2.29E+01+	3.47E+01+	9.91E+01+	4.43E+01+	1.74E+01+	**2.98E+01**
F28	mean	3.21E+02+	1.76E+02+	8.17E+02+	2.03E+02+	2.07E+02+	**1.65E+02**
	std	4.01E+01+	2.69E+01+	8.33E+01+	4.11E+01+	5.43E+01+	**1.59E+01**
F29	mean	5.37E+01+	3.57E+01+	6.48E+01+	3.36E+01+	**2.95E+01**-	3.31E+01
	std	1.14E+00+	8.05E+00+	2.74E+00+	4.66E+00+	**7.07E+00**-	4.33E+00
F30	mean	**4.00E-01**-	5.33E-01-	2.27E+00+	1.33E+00+	4.33E-01-	6.25E-01
	std	**2.99E-08**-	1.16E-01-	3.51E-01+	1.53E-01+	5.77E-02-	9.57E-02
	- / ≈ / +	20/4/6	23/2/5	27/0/3	25/2/3	23/1/6	

It can be seen that PSO-HLM not only has the best solution result on F1, F2, F3, F5 and F6 functions, but also has the fastest convergence speed. Among other unimodal functions F4 and F8, PSO-HLM also has the fastest convergence speed or the best solution accuracy. This strongly shows that increasing early-stage diversity can help generate good particles and promote the algorithm for finding the best solution. On F7, PSO-HLM falls into local optima at the early stage, but escapes from local optima with the help of four hybrid learning model strategy. At the same time, it also shows that in different learning models and Gaussian disturbances, it can avoid falling into the local optima prematurely due to the sharp decrease of diversity. On F10, F11 and F12 of basic multimodal functions, PSO-HLM has the best performance in finding the global solution. On F9 and F14, PSO-HLM has the fastest convergence speed, although it does not have a good solution accuracy. It can also be seen from F15 that diversified learning models can provide more possibilities for later algorithms.

Among multimodal functions, PSO-HLM has overwhelmed advantages in many functions, including basic functions and modified functions. It can be seen from that PSO-HLM has good convergence speed and search speed for many functions, and has relatively optimal solutions for most functions. However, on F16, F18, and F19, it still has much locally optimal, possibly because the better learning model or learning sample has not been found. But on F21 and F22, PSO-HLM has an overwhelming advantage performance. The convergence curves of these functions F20, F23, F26 is shown that different types of learning models were used in the middle and later stages, which kept a balance between local search and global search. It is allowed to search within an appropriate range because it avoids local optimization. On other modified multimodal functions, PSO-HLM has the best performance on F27 and F28, but it fails to escape the local optima at F25 and F29. This suggests that in some cases there is a need to improve the division of multiple learning patterns.

4.3.3. Analysis of experimental results on 50-D

From Table 5, PSO-HLM still has the best performance on high-dimensional optimization problems. On unimodal functions, PSO-HLM has the best fitness mean values. The results of F3, F6 and F8 functions illustrate the unique merits of PSO-HLM in such problems, PSO-HLM can fully explore the best solutions by using alternative updates of these four learning models. On F4 function, the results for XPSO are better than PSO-HLM, it indicated that the population forgetting capability is also optimal for unimodal functions to avoid falling into local optimal solution. F11, F12 and F19 functions have good solutions for PSO-HLM, TAPSO and CLPSO, while the results are poor for XPSO and PPSO. It confirms that the learning model has a great influence on the search for problem solutions, it indicated that a more suitable learning model can facilitate the solution exploration well. On F13, F15 functions, the experimental results of CLPSO are the best, and only CLPSO has found the optimal solution for F13. This shows that when solving this kind of function, it is very important to use the learning model to update particles into the best samples. For the modified multimodal functions, PSO-HLM has the first optimal solution proportion, then CLPSO, TAPSO, XPSO and PPSO, respectively. The comparison of TAPSO and PSO-HLM illustrates that for different improvement rates can effectively help to learn the model turnover and locate the optimal solution more precisely. The results of F25, F28 functions illustrate that for the solution accuracy of this problem PSO-HLM has a significant improvement, it proved that the combination of the improvement rate of PSO-HLM and the selection of multiple learning models can effectively improve the efficiency of PSO-HLM.

4.3.4. Analysis of experimental results on CEC 2017 30-D

From Table 6, PSO-HLM is compared with other algorithms on the CEC2017 test functions. On unimodal function (F1), an excellent CLPSO performance is achieved. It shows that the stable learning model has a good advantage for this type of problem. On simple multimodal function (F2–F8), PSO-HLM exemplifies that the multi-learning model is the best choice for jumping out of the local optimal solution. On hybrid function (F9–F18), CLPSO and PSO-HLM have equal and maximum number of optimal solutions, and the combination of crossover probabilities under adaptive and multiple learning models can explore optimal solutions as effectively as CLPSO. On composition function (F19–F28), These algorithms (CLPSO, TAPSO, PSO-HLM, XPSO, PPSO, and FDR-PSO) have the same results. This shows that there is not much difference between fixed learning model and adaptive learning model for such problems.

4.4. Effectiveness analysis

4.4.1. Effectiveness of the adaptive parameters strategy

The inertia weights and crossover probability plays a key role in PSO-HLM. In order to fully test the performance of the adaptive parameters strategy, two algorithms (PSO-HLM-NA and PSO-HLM-A) are compared on the benchmark function suite, where PSO-HLM-NA uses fixed parameters(w = 0.7298, p_c =0.5, followed by TAPSO [18]), and PSO-HLM-A uses adaptive parameter strategy. All experimental results are shown in Table 7.

From Table 7, PSO-HLM-A significantly outperforms PSO-HLM-NA. Because, in the adaptive parameter strategy, the inertia weights and crossover probability using the triangular function, the center of gravity of each stage can be fully utilized. Large inertia weight can improve the ability of exploration, and small inertia weight can improve the ability of exploitation. In the iterative process, These four learning models can adaptively adjust the parameters, the adaptive inertia weight can make the particles fit well with different behaviors. The center of gravity in the stage is also controlled iteratively by the sinusoidal function to realize that it is more likely to escape the local optimization in the later stage. It can be found that PSO-HLM-A is more likely to jump out of the local optima and find a more potential solution in the later stage.

4.4.2. Effectiveness of the multi-pool fusion strategy and hybrid learning model strategy

From Table 8, it can be seen that PSO-HLM-F has better performance than PSO-HLM-NF. In the original algorithm, the useful information of sub-excellent particles cannot be well utilized, and many promising possibilities were lost. The proposed multi-pool fusion strategy, based on two points: 1) one is the characteristics of different subpopulation pools, and 2) another is the information of multiple parents. It enables PSO-HLM-F introduce more possibilities and also avoids falling into local optima to some extent in the late stage, the exemplar generated by the multi-pool fusion strategy can fuse the characteristics of different subpopulation pools.

From Table 9, PSO-HLM-H fully outperforms PSO-HLM-NH, where PSO-HLM-H uses hybrid learning model strategy, and PSO-HLM-NH only uses standard learning model strategy. In the hybrid learning model strategy, these four learning model (confidence learning model, mild learning model, standard learning model, Gaussian learning model) can make full use of the total information of particles at each stage, so that each particles can absorb various effective information. At the stage of sharp decline in diversity, the transformation between different learning models is realized by implementing Gaussian perturbation, thus it can avoid the local concentration of particles caused by a single learning model. Gauss model and standard model can further enhance the randomness and regularity of the learning stage.

Table 6. The algorithm results for CEC2017 30-D.

No.		CLPSO	FDR-PSO	PPSO	TAPSO	XPSO	PSO-HLM
F1	mean	**1.01E+02**-	4.93E+03 +	1.11E+04 +	5.28E+02 +	4.93E+03 +	3.36E+02
	std	**5.12E-01**-	8.27E+03 +	9.65E+03 +	7.17E+02 +	3.94E+03 +	2.33E+02
F2	mean	1.83E+04 +	**3.00E+02**≈	3.00E+02 +	**3.00E+02**≈	3.00E+02 +	**3.00E+02**
	std	2.46E+03 +	**0.00E+00**≈	1.05E-04 +	**0.00E+00**≈	7.96E-04 +	**0.00E+00**
F3	mean	4.90E+02 +	4.30E+02 +	4.97E+02 +	4.23E+02 +	5.34E+02 +	**4.03E+02**
	std	1.39E+01 +	4.58E+01 +	1.39E+01 +	3.59E+01 +	1.23E+01 +	**2.23E+00**
F4	mean	5.53E+02 +	5.48E+02 +	7.60E+02 +	5.48E+02 +	5.45E+02 +	**5.41E+02**
	std	7.54E+00 +	4.34E+00 +	1.45E+01 +	1.15E+01 +	1.29E+01 +	**4.17E+00**
F5	mean	**6.00E+02**-	6.00E+02 +	6.35E+02 +	6.00E+02 -	6.00E+02 +	6.00E+02
	std	**0.00E+00**-	5.40E-03 +	1.12E+01 +	1.80E-03 -	5.45E-01 +	3.00E-03
F6	mean	7.87E+02 +	7.92E+02 +	1.12E+03 +	7.74E+02 +	7.72E+02 +	**7.67E+02**
	std	6.43E+00 +	1.80E+01 +	3.22E+01 +	1.41E+01 +	3.19E+00 +	**4.94E+00**
F7	mean	8.49E+02 +	8.47E+02 +	9.79E+02 +	8.52E+02 +	8.45E+02 +	**8.38E+02**
	std	1.03E+01 +	4.14E+00 +	2.19E+01 +	3.20E+00 +	6.05E+00 +	**3.04E+00**
F8	mean	9.09E+02 -	9.05E+02 -	5.12E+03 +	9.88E+02 +	**9.04E+02** -	9.10E+02
	std	3.05E+00 -	4.44E+00 -	6.91E+02 +	1.01E+02 +	**2.11E+00** -	4.31E+00
F9	mean	**3.27E+03** -	3.85E+03 +	5.39E+03 +	3.45E+03-	3.91E+03 +	3.52E+03
	std	**2.89E+02** -	8.63E+02 +	7.98E+02 +	4.37E+02-	7.12E+02 +	2.98E+02
F10	mean	1.18E+03 +	1.23E+03 +	1.21E+03 +	1.18E+03 +	1.19E+03 +	**1.15E+03**
	std	4.51E+00 +	6.23E+01 +	7.41E+00 +	1.81E+01 +	5.34E+01 +	**2.02E+01**
F11	mean	5.97E+05 +	1.89E+04 +	2.53E+05 +	2.37E+04 +	1.44E+05 +	**1.48E+04**
	std	3.96E+05 +	1.08E+04 +	2.95E+05 +	1.07E+04 +	1.45E+05 +	**1.61E+03**
F12	mean	**2.62E+03**-	1.97E+04+	3.81E+03-	2.71E+04+	5.89E+03-	6.06E+03
	std	**1.55E+03**-	3.58E+03+	1.19E+03-	2.93E+04+	4.17E+03-	5.31E+03
F13	mean	1.75E+04 +	3.20E+03 +	**1.53E+03**-	1.92E+03-	4.53E+03 +	2.52E+03
	std	9.65E+03 +	1.55E+03 +	**5.23E+01**-	5.05E+02-	2.47E+03 +	5.95E+02
F14	mean	**1.61E+03**-	2.01E+03-	1.68E+03-	2.28E+03-	1.72E+04+	5.89E+03
	std	**2.70E+01**-	3.70E+02-	7.78E+01-	5.13E+02-	1.52E+04+	4.95E+03
F15	mean	2.16E+03+	2.18E+03+	2.84E+03+	2.48E+03+	2.44E+03+	**2.13E+03**
	std	1.32E+02+	4.88E+01+	4.40E+02+	2.01E+02+	1.03E+02+	**2.25E+02**
F16	mean	1.84E+03 +	1.82E+03 +	2.34E+03 +	1.96E+03 +	1.89E+03 +	**1.79E+03**
	std	4.18E+01 +	1.01E+02 +	4.10E+02 +	5.75E+01 +	1.56E+02 +	**6.86E+01**
F17	mean	2.16E+05 +	9.21E+04 +	**4.71E+03**-	5.76E+04+	3.02E+05+	3.25E+04
	std	1.61E+05 +	8.01E+04 +	**2.50E+03**-	1.81E+04+	3.22E+05+	3.87E+03
F18	mean	**1.95E+03**-	5.34E+03 +	2.10E+03-	6.30E+03 +	1.66E+04 +	5.24E+03
	std	**1.88E+01**-	3.21E+03 +	1.30E+02-	1.69E+03 +	1.82E+04 +	4.25E+03
F19	mean	2.20E+03+	2.24E+03+	2.51E+03+	2.18E+03-	**2.14E+03**-	2.19E+03
	std	2.64E+01+	1.10E+02+	2.04E+02+	1.20E+01-	**8.24E+01**-	1.27E+02
F20	mean	2.36E+03+	2.35E+03+	2.53E+03+	2.34E+03+	2.35E+03+	**2.34E+03**
	std	5.04E+00+	8.88E+00+	4.82E+01+	1.24E+01+	1.10E+01+	**1.39E+01**
F21	mean	2.65E+03-	3.39E+03+	3.48E+03+	**2.30E+03**-	2.30E+03-	2.98E+03
	std	5.40E+02-	1.89E+03+	2.04E+03+	**1.97E+00**-	2.10E+00-	1.17E+03
F22	mean	2.72E+03+	2.71E+03+	2.98E+03+	2.71E+03+	**2.68E+03**-	2.71E+03
	std	1.15E+01+	1.79E+01+	6.70E+01+	1.47E+01+	**5.30E+00**-	8.36E+00
F23	mean	2.92E+03+	2.88E+03+	3.21E+03+	2.89E+03+	2.91E+03+	**2.88E+03**
	std	1.21E+01+	2.22E+01+	2.00E+01+	1.37E+01	4.41E+01+	**1.69E+01**
F24	mean	2.89E+03-	2.89E+03-	2.92E+03+	**2.89E+03**-	2.90E+03+	2.89E+03
	std	2.23E-01-	5.30E-01-	2.54E+01+	**2.62E+00**-	1.58E+01+	4.20E+00
F25	mean	**3.47E+03**-	4.30E+03+	5.24E+03+	5.20E+03+	3.82E+03-	3.87E+03
	std	**7.89E+02**-	3.81E+02+	2.15E+03+	1.03E+03+	7.96E+02-	9.38E+02
F26	mean	**3.21E+03**-	3.22E+03+	3.30E+03+	3.23E+03+	3.23E+03+	3.22E+03
	std	**6.70E+00**-	1.31E+01+	5.53E+01+	6.71E+00+	2.87E+01+	2.14E+00
F27	mean	3.21E+03+	3.15E+03+	3.24E+03+	**3.10E+03**-	3.18E+03+	3.13E+03
	std	5.65E+00+	5.19E+01+	2.88E+01+	**0.00E+00**-	7.61E+01+	5.96E+01
F28	mean	**3.45E+03**-	3.47E+03+	3.99E+03+	3.75E+03+	3.49E+03-	3.54E+03
	std	**4.85E+00**-	7.35E+01+	2.63E+02+	2.18E+02+	1.06E+02-	1.65E+02
	- / ≈ / +	12/0/16	24/1/3	23/0/5	19/1/8	21/0/7	

Table 7. The results of the parameter adaptation experiments, where PSO-HLM-NA uses fixed parameters, and PSO-HLM-A uses adaptive parameter strategy.

No.		PSO-HLM-NA	PSO-HLM-A
F1	mean	1.94E-160 -	**7.61E-175**
	std	6.24E-160 -	**0.00E+00**
F2	mean	1.34E-78 -	**9.30E-89**
	std	4.54E-78 -	**2.19E-88**
F3	mean	2.21E-26 -	**1.95E-31**
	std	7.95E-26 -	**4.86E-31**
F4	mean	3.91E+03 -	**3.60E+03**
	std	1.15E+03 -	**7.50E+02**
F5	mean	**0.00E+00** ≈	**0.00E+00**
	std	**0.00E+00** ≈	**0.00E+00**
F6	mean	6.46E-20 -	**2.12E-21**
	std	3.50E-19 -	**8.07E-21**
F7	mean	**8.69E+03** +	1.01E+04
	std	**4.07E+03** +	3.88E+03
F8	mean	2.23E+05 -	**1.59E+05**
	std	1.17E+05 -	**1.21E+05**
F9	mean	**5.57E-15** +	6.51E-15
	std	**1.79E-15** +	1.35E-15
F10	mean	1.82E-13 +	**0.00E+00**
	std	5.55E-13 +	**0.00E+00**
F11	mean	**0.00E+00** ≈	**0.00E+00**
	std	**0.00E+00** ≈	**0.00E+00**
F12	mean	**0.00E+00** ≈	**0.00E+00**
	std	**0.00E+00** ≈	**0.00E+00**
F13	mean	2.83E-02 -	**2.13E-02**
	std	4.31E-02 -	**2.31E-02**
F14	mean	**1.59E-32** +	1.61E-32
	std	**9.43E-34** +	1.31E-33
F15	mean	**5.83E-01** +	6.07E-01
	std	**1.39E-01** +	1.39E-01
	-/ ≈ / +	8/3/4	

No.		PSO-HLM-NA	PSO-HLM-A
F16	mean	**3.38E+00** +	3.52E+00
	std	**7.82E-01** +	7.20E-01
F17	mean	5.22E-12 -	**2.64E-12**
	std	8.08E-12 -	**1.36E-11**
F18	mean	5.90E-03 -	**5.70E-03**
	std	8.30E-03 -	**6.80E-03**
F19	mean	**1.14E+00** +	1.17E+00
	std	**2.47E-01** +	1.83E-01
F20	mean	2.17E+03 -	**1.64E+03**
	std	4.32E+03 -	**2.69E+03**
F21	mean	**0.00E+00** ≈	**0.00E+00**
	std	**0.00E+00** ≈	**0.00E+00**
F22	mean	**0.00E+00** ≈	**0.00E+00**
	std	**0.00E+00** ≈	**0.00E+00**
F23	mean	5.47E-04 -	**2.28E-06**
	std	2.40E-03 -	**1.09E-05**
F24	mean	**1.20E+01** +	1.21E+01
	std	**5.90E-01** +	6.95E-01
F25	mean	**1.82E-02** +	2.22E-02
	std	**1.83E-02** +	3.07E-02
F26	mean	**2.03E+01** ≈	**2.03E+01**
	std	**4.01E-01** ≈	**4.34E-01**
F27	mean	7.34E+01 -	**6.69E+01**
	std	1.88E+01 -	**1.94E+01**
F28	mean	9.00E+01 -	**8.41E+01**
	std	2.81E+01 -	**2.39E+01**
F29	mean	1.99E+01 -	**1.74E+01**
	std	3.56E+00 -	**3.05E+00**
F30	mean	**5.43E-01** +	6.20E-01
	std	**1.36E-01** +	1.22E-01
	+/ ≈ / +	7/3/5	

Table 8. The effectiveness experimental results of the multi-pool fusion strategy, where PSO-HLM-NF uses a total population, and PSO-HLM-F uses multi-pool subpopulation fusion strategy.

No.		PSO-HLM-NF	PSO-HLM-F
F1	mean	1.94E-160 -	**7.81E-188**
	std	6.24E-160 -	**0.00E+00**
F2	mean	1.34E-78 -	**1.45E-96**
	std	4.54E-78 -	**4.67E-96**
F3	mean	2.21E-26 -	**4.45E-33**
	std	7.95E-26 -	**1.75E-32**
F4	mean	3.91E+03 -	**3.15E+03**
	std	1.15E+03 -	**5.59E+02**
F5	mean	**0.00E+00** ≈	**0.00E+00**
	std	**0.00E+00** ≈	**0.00E+00**
F6	mean	6.46E-20 -	**7.32E-27**
	std	3.50E-19 -	**2.63E-27**
F7	mean	8.69E+03 -	**7.12E+03**
	std	4.07E+03 -	**2.20E+03**
F8	mean	2.23E+05 -	**1.11E+05**
	std	1.17E+05 -	**4.48E+04**
F9	mean	**5.57E-15** +	6.28E-15
	std	**1.79E-15** +	1.53E-15
F10	mean	1.82E-13 -	**0.00E+00**
	std	5.55E-13 -	**0.00E+00**
F11	mean	**0.00E+00** ≈	**0.00E+00**
	std	**0.00E+00** ≈	**0.00E+00**
F12	mean	**0.00E+00** ≈	**0.00E+00**
	std	**0.00E+00** ≈	**0.00E+00**
F13	mean	2.83E-02 -	**9.10E-03**
	std	4.31E-02 -	**7.50E-03**
F14	mean	**1.59E-32** +	1.61E-32
	std	**9.43E-34** +	1.31E-33
F15	mean	5.83E-01 -	**5.20E-01**
	std	1.39E-01 -	**8.05E-02**
	-/ ≈ / +	10/3/2	

No.		PSO-HLM-NF	PSO-HLM-F
F16	mean	**3.38E+00** +	3.42E+00
	std	**7.82E-01** +	4.98E-01
F17	mean	5.22E-12 -	**2.04E-14**
	std	8.08E-12 -	**6.40E-14**
F18	mean	5.90E-03 -	**3.70E-03**
	std	8.30E-03 -	**6.10E-03**
F19	mean	1.14E+00 -	**1.11E+00**
	std	2.47E-01 -	**2.30E-01**
F20	mean	2.17E+03 -	**8.87E+02**
	std	4.32E+03 -	**1.03E+03**
F21	mean	**0.00E+00** ≈	**0.00E+00**
	std	**0.00E+00** ≈	**0.00E+00**
F22	mean	**0.00E+00** ≈	**0.00E+00**
	std	**0.00E+00** ≈	**0.00E+00**
F23	mean	5.47E-04 -	**3.20E-10**
	std	2.40E-03 -	**1.38E-09**
F24	mean	**1.20E+01** +	1.21E+01
	std	**5.90E-01** +	5.43E-01
F25	mean	**1.82E-02** +	1.84E-02
	std	**1.83E-02** +	1.83E-02
F26	mean	**2.03E+01** ≈	**2.03E+01**
	std	**4.01E-01** ≈	**4.03E-01**
F27	mean	7.34E+01 -	**6.33E+01**
	std	1.88E+01 -	**1.40E+01**
F28	mean	9.00E+01 -	**7.50E+01**
	std	2.81E+01 -	**2.10E+01**
F29	mean	1.99E+01 -	**1.83E+01**
	std	3.56E+00 -	**2.56E+00**
F30	mean	**5.43E-01** +	6.03E-01
	std	**1.36E-01** +	1.30E-01
	-/ ≈ / +	8/3/4	

Table 9. The effectiveness experimental results of four hybrid model strategy. where PSO-HLM-NH only uses standard learning model strategy, and PSO-HLM-H uses hybrid learning model strategy.

No.		PSO-HLM-NH	PSO-HLM-H	No.		PSO-HLM-NH	PSO-HLM-H
F1	mean	1.94E-160 -	**8.23E-164**	F16	mean	**3.38E+00** +	3.69E+00
	std	6.24E-160 -	**0.00E+00**		std	**7.82E-01** +	8.37E-01
F2	mean	1.34E-78 -	**9.96E-80**	F17	mean	**5.22E-12** +	2.32E-10
	std	4.54E-78 -	**3.22E-79**		std	**8.08E-12** +	6.90E-10
F3	mean	2.21E-26 -	**3.94E-29**	F18	mean	5.90E-03 -	**4.80E-03**
	std	7.95E-26 -	**2.03E-28**		std	8.30E-03 -	**7.80E-03**
F4	mean	3.91E+03 -	**3.72E+03**	F19	mean	1.14E+00 -	**1.13E+00**
	std	1.15E+03 -	**9.66E+02**		std	2.47E-01 -	**2.07E-01**
F5	mean	**0.00E+00** ≈	**0.00E+00**	F20	mean	2.17E+03 -	**1.90E+03**
	std	**0.00E+00** ≈	**0.00E+00**		std	4.32E+03 -	**2.87E+03**
F6	mean	6.46E-20 -	**8.30E-27**	F21	mean	**0.00E+00** ≈	**0.00E+00**
	std	3.50E-19 -	**2.16E-26**		std	**0.00E+00** ≈	**0.00E+00**
F7	mean	8.69E+03 -	**6.95E+02**	F22	mean	**0.00E+00** ≈	**0.00E+00**
	std	4.07E+03 -	**4.42E+02**		std	**0.00E+00** ≈	**0.00E+00**
F8	mean	2.23E+05 -	**1.80E+05**	F23	mean	5.47E-04 -	**8.58E-07**
	std	1.17E+05 -	**1.06E+05**		std	2.40E-03 -	**4.25E-06**
F9	mean	**5.57E-15** +	6.63E-15	F24	mean	1.20E+01 -	**1.19E+01**
	std	**1.79E-15** +	1.23E-15		std	5.90E-01 -	**6.70E-01**
F10	mean	1.82E-13 -	**0.00E+00**	F25	mean	**1.82E-02** +	1.86E-02
	std	5.55E-13 -	**0.00E+00**		std	**1.83E-02** +	1.74E-02
F11	mean	**0.00E+00** ≈	**0.00E+00**	F26	mean	**2.03E+01** ≈	**2.03E+01**
	std	**0.00E+00** ≈	**0.00E+00**		std	**4.01E-01** ≈	**4.25E-01**
F12	mean	**0.00E+00** ≈	**0.00E+00**	F27	mean	7.34E+01 -	**5.89E+01**
	std	**0.00E+00** ≈	**0.00E+00**		std	1.88E+01 -	**1.23E+01**
F13	mean	2.83E-02 -	**1.00E-03**	F28	mean	9.00E+01 -	**7.01E+01**
	std	4.31E-02 -	**2.00E-03**		std	2.81E+01 -	**1.74E+01**
F14	mean	**1.59E-32** +	1.61E-32	F29	mean	1.99E+01 -	**1.75E+01**
	std	**9.43E-34** +	1.31E-33		std	3.56E+00 -	**2.75E+01**
F15	mean	5.83E-01 -	**4.67E-01**	F30	mean	5.43E-01 -	**4.57E-01**
	std	1.39E-01 -	**6.61E-02**		std	1.36E-01 -	**8.58E-02**
	-/ ≈ / +	10/3/2			-/ ≈ / +	9/3/3	

Table 10. The sensitivity experimental results of hybrid learning model strategy initialization time.

No.		0	MaxIter/3	2*MaxIter/3
F1	mean	6.90E-172 -	3.62E-180	**8.80E-188**-
	std	0.00E+00 -	0.00E+00	**0.00E+00** +
F2	mean	3.29E-84 -	3.81E-91	**1.49E-96**-
	std	1.67E-83 -	1.39E-90	**5.99E-96**-
F3	mean	1.52E-29 -	1.17E-31	**3.56E-33**-
	std	3.61E-29 -	2.97E-31	**1.77E-32**-
F4	mean	4.37E+03 -	**3.41E+03**	4.05E+03+
	std	8.46E+02 -	**9.02E+02**	1.01E+03+
F5	mean	**0.00E+00** ≈	**0.00E+00**	**0.00E+00** ≈
	std	**0.00E+00** ≈	**0.00E+00**	**0.00E+00** ≈
F6	mean	**4.12E-27** +	6.59E-27	7.58E-18+
	std	**2.12E-27** +	4.96E-27	4.15E-17+
F7	mean	**4.13E+02** +	1.13E+03	1.01E+04+
	std	**2.29E+02** +	8.29E+02	3.71E+03+
F8	mean	2.05E+05 -	1.67E+05	**1.41E+05**-
	std	9.96E+04 -	7.43E+04	**7.08E+04**-
F9	mean	6.87E-15 -	**6.63E-15**	6.75E-15+
	std	9.01E-16 -	**1.23E-15**	1.08E-15+
F10	mean	2.43E-13 -	**0.00E+00**	4.85E-13+
	std	6.29E-13 -	**0.00E+00**	8.18E-13+
F11	mean	**0.00E+00** ≈	**0.00E+00**	**0.00E+00** ≈
	std	**0.00E+00** ≈	**0.00E+00**	**0.00E+00** ≈
F12	mean	**0.00E+00** ≈	**0.00E+00**	**0.00E+00** ≈
	std	**0.00E+00** ≈	**0.00E+00**	**0.00E+00** ≈
F13	mean	**3.74E-04** +	8.96E-04	2.85E-02+
	std	**3.69E-04** +	1.50E-03	3.03E-02+
F14	mean	**1.57E-32** +	1.59E-32	1.61E-32 -
	std	**5.57E-46** +	9.43E-34	1.31E-33 -
F15	mean	**3.93E-01** +	4.80E-01	5.73E-01 -
	std	**6.91E-02** +	7.61E-02	1.14E-01 -
	-/ ≈ / +	7/3/5		8/3/4

No.		0	MaxIter/3	2*MaxIter/3
F16	mean	3.98E+00 -	**3.64E+00**	3.69E+00+
	std	6.27E-01 -	**7.33E-01**	6.97E-01 -
F17	mean	1.18E-09 -	3.53E-12	**3.60E-14**-
	std	2.20E-09 -	5.04E-12	**8.85E-14**-
F18	mean	8.12E-03 -	**5.10E-03**	5.60E-03+
	std	1.05E-02 -	**1.02E-02**	1.01E-02+
F19	mean	1.18E+00 -	**1.12E+00**	**1.12E+00** ≈
	std	2.33E-01 -	**1.78E-01**	**2.00E-01** ≈
F20	mean	3.37E+03 -	**9.95E+02**	2.10E+03+
	std	5.34E+03 -	**1.40E+03**	3.53E+03+
F21	mean	**0.00E+00** ≈	**0.00E+00**	**0.00E+00** ≈
	std	**0.00E+00** ≈	**0.00E+00**	**0.00E+00** ≈
F22	mean	**0.00E+00** ≈	**0.00E+00**	**0.00E+00** ≈
	std	**0.00E+00** ≈	**0.00E+00**	**0.00E+00** ≈
F23	mean	3.39E-01 -	**4.28E-10**	5.30E-10+
	std	1.31E+00 -	**1.45E-09**	1.77E-09+
F24	mean	1.23E+01 -	**1.21E+01**	1.22E+01+
	std	4.14E-01 -	**6.25E-01**	8.04E-01+
F25	mean	**1.44E-02** +	2.53E-02	2.12E-02-
	std	**1.25E-02** +	3.31E-02	3.23E-02-
F26	mean	2.09E+01 -	2.04E+01	**2.02E+01**-
	std	5.29E-02 -	4.65E-01	**3.59E-01** +
F27	mean	7.39E+01 -	**5.46E+01**	5.57E+01+
	std	4.34E+01 -	**1.51E+01**	1.50E+01+
F28	mean	8.25E+01 -	**7.10E+01**	7.67E+01+
	std	3.28E+01 -	**1.71E+01**	2.63E+01+
F29	mean	**1.66E+01** +	1.70E+01	1.72E+01+
	std	**2.20E+00** +	2.49E+00	2.92E+00+
F30	mean	**4.03E-01** +	4.80E-01	5.70E-01+
	std	**7.18E-02** +	7.13E-02	9.15E-02+
	-/ ≈ / +	10/2/3		9/3/3

4.5. Sensitivity analysis

In order to ensure the sensitivity of the hybrid learning model strategy, several experimental analysis was carried out to ensure that the initialization time of the hybrid learning model strategy. The experimental result was shown on Table 10, where 0, $MaxIter/3$, and $2 * MaxIter/3$ represents the hybrid learning model strategy initialization time. Due to the introduction of multi-pool fusion strategy and different learning models, PSO-HLM can be divided into three stages: prospect, transition and development.

From Table 10, it can be see that the initialization time of the hybrid learning model strategy equal to $MaxIter/3$, PSO-HLM has the best performance than other two initialization time (0 and $2 * MaxIter/3$). Because, in the early stage of iteration, the initialized population has a strong search ability. In the medium stage ($MaxIter/3$), the diversity of the population decreases, and these four learning strategies can effectively improve the exploration ability of the population. In the last stage ($2 * MaxIter/3$), the computing resources are not enough for the population to explore new landscape.

Thus, $MaxIter/3$ is the best time to start hybrid learning model strategy.

5. Conclusions

In this paper, a hybrid learning model based particle swarm optimization algorithm (PSO-HLM) is proposed. These improved PSO-HLM strategies can increase population diversity and avoid falling into local optimization. In PSO-HLM, the multi-pool fusion strategy can increase population diversity and avoid sudden decline of population diversity. In addition, in the particle learning stage, each particle can update its velocity through four learning models. Confidence learning model makes particles search for sub-particles with the best potential exemplar. Mild learning model enable particles to learn the most suitable particle from two different sub-populations (the potential exemplar and global optimal particles), thus it can balance multi-directional search. Standard learning model enable particles to search in multiple directions by learning global optimal particles and local optimal particles. Gaussian model makes particles jump out of the local optimal solution with Gaussian perturbation. These four learning models can ensure that PSO-HLM can fully learn the information of each particle and avoid falling into local optimization prematurely.

In PSO-HLM, these different learning models can help particles find more potential solution, and avoid falling into local optimization prematurely. In future work, the dynamic changes of population size under different learning models can be further studied.

Acknowledgements

This work was supported in part by the Key Research Projects of Henan Science and Technology Department under Grant 212102210492, and in part by the Henan Science and Technology Think Tank Research Project under Grant HNKJZK-2023-51B.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. G. G. Wang, S. Deb, Z. Cui, Monarch butterfly optimization, *Neural Comput. Appl.*, **31** (2019), 1995–2014. https://doi.org/10.1007/s00521-015-1923-y
2. S. Li, H. Chen, M. Wang, A. A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, *Future Gener. Comput. Syst.*, **111** (2020), 300–323. https://doi.org/10.1016/j.future.2020.03.055
3. C. G. Wang, Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems, *Memetic Comput.*, **10** (2018), 151–164. https://doi.org/10.1007/s12293-016-0212-3
4. Y. Yang, H. Chen, A. A. Heidari, A. Gandomi, Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts, *Expert Syst. Appl.*, **177** (2021), 114864.

5. J. C. Butcher, A history of Runge-Kutta methods, *Appl. Numer. Math.*, **20** (1996), 247–260. https://doi.org/10.1016/0168-9274(95)00108-5
6. J. Tu, H. Chen, M. Wang, A. H. Gandomi, The colony predation algorithm, *J. Bionic Eng.*, **18** (2021), 674–710.
7. I. Ahmadianfar, A. A. Heidari, S. Noshadian, H. Chen, A. H. Gandomi, INFO: An efficient optimization algorithm based on weighted mean of vectors, *Expert Syst. Appl.*, **195** (2022), 116516. https://doi.org/10.1016/j.eswa.2022.116516
8. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, *Future Gener. Comput. Syst.*, **97** (2019), 849–872. https://doi.org/10.1016/j.future.2019.02.028
9. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of ICNN'95-international conference on neural networks*, (1995), 1942–1948.
10. X. J. Yang, Q. J. Jiao, X. Liu, Center particle swarm optimization algorithm, in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, (2019), 2084–2087.
11. B. Wei, X. Xia, F. Yu, Y. Zhang, X. Xu, H. Wu, et al., Multiple adaptive strategies based particle swarm optimization algorithm, *Swarm Evol. Comput.*, **57** (2020), 100731. https://doi.org/10.1016/j.swevo.2020.100731
12. N. Lynn, P. N. Suganthan, Ensemble particle swarm optimizer, *Appl. Soft Comput.*, **55** (2017), 533–548. https://doi.org/10.1016/j.asoc.2017.02.007
13. Cao H, Zheng H, Hu G. Generation of quasi-developable Q-Bezier strip via PSO-based shape parameters optimization, *Math. Methods Appl. Sci.*, **45** (2022), 1118–1129. https://doi.org/10.1002/mma.7839
14. Q. Cui, C. Tang, G. Xu, C. Wu, X. Shi, Y. Liang, et al., Surprisingly popular algorithm-based comprehensive adaptive topology learning PSO, in *2019 IEEE Congress on Evolutionary Computation (CEC)*, (2019), 2603–2610. https://doi.org/10.1109/CEC.2019.8790002
15. A. P. Engelbrecht, B. S. Masiye, G. Pampard, Niching ability of basic particle swarm optimization algorithms, in *Proceedings 2005 IEEE Swarm Intelligence Symposium*, (2005), 397–400.
16. Dong H, Zhang H, Han S, X. Li, X. Wang, Reverse-learning particle swarm optimization algorithm based on niching technology, in *2018 IEEE/ACIS 17th international conference on computer and information science (ICIS)*, (2018), 405–410.
17. Z. Du, S. Li, Y. Sun Y, N. Li, Adaptive particle swarm optimization algorithm based on levy flights mechanism, in *2017 Chinese Automation Congress (CAC)*, (2017), 479–484.
18. X. Xia, L. Gui, F. Yu, H. Wu, B. Wei, Y. Zhang, et al., Triple archives particle swarm optimization, *IEEE Trans. Cybern.*, **50** (2019), 4862–4875. https://doi.org/10.1109/TCYB.2019.2943928
19. D. K. Kole, A. Halder, An efficient dynamic image segmentation algorithm using a hybrid technique based on particle swarm optimization and genetic algorithm, in *2010 International Conference on Advances in Computer Engineering*, (2010), 252–255.
20. A. Colorni, M. Dorigo, V. Maniezzo, Distributed optimization by ant colonies. *Proceedings of the First European Conference on Artificial Life*, (1998), 134–142.

21. R. C. Eberhart, J. Kennedy, A new optimiser using particle swarm theory, in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, IEEE, (1995), 39–43.
22. Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, *IEEE Trans. Evol. Comput.*, **10** (2006), 281–295. https://doi.org/10.1109/TEVC.2005.857610
23. O. M. Sedeh, B. Ostadi, F. Zagia, A novel hybrid GA-PSO optimization technique for multi-location facility maintenance scheduling problem, *J. Build. Eng.*, **40** (2021), 102348.
24. M. Gao, Y. Zhu, J. Sun, The multi-objective cloud tasks scheduling based on hybrid particle swarm optimization, in *2020 Eighth International Conference on Advanced Cloud and Big Data (CBD)*, (2020).
25. X. Liu, H. Yi, Z. Ni, Application of ant colony optimization algorithm in process planning optimization, in *2012 Fifth International Conference on Intelligent Networks and Intelligent Systems*, (2012), 61–64. https://doi.org/10.1111/j.1365-2559.2012.04359_7.x
26. S. Kefi, N. Rokbani, P. Kromer, A. M. Alimi, Ant supervised by PSO and 2-opt algorithm, AS-PSO-2Opt, applied to traveling salesman problem, in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, (2016), 4866–4871. https://doi.org/10.1109/SMC.2016.7844999
27. J. Lu, J. Zhang, J. Sheng, Enhanced multi-swarm cooperative particle swarm optimizer, *Swarm Evol. Comput.*, **69** (2022), 100989. https://doi.org/10.1016/j.swevo.2021.100989
28. Y. Wang, W. Dong, X. Dong, Cooperative Coevolution with Correlation Learning Between Variables for Large Scale Overlapping Problem, *Acta Electon. Sin.*, **48** (2018), 529.
29. Y. Gao, An improved hybrid group intelligent algorithm based on artificial bee colony and particle swarm optimization, in *2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*, (2018), 160–163.
30. O. Ramos-Figueroa, M. Quiroz-Castellanos, E. Mezura-Montes, R. Kharel, Variation operators for grouping genetic algorithms: A review, *Swarm Evol. Comput.*, **60** (2021), 100796. https://doi.org/10.1016/j.swevo.2020.100796
31. A. Ratnaweera, S. K. Halgamuge, H. C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, *IEEE Trans. Evol. Comput.*, **8** (2004), 240–255. https://doi.org/10.1109/TEVC.2004.826071
32. T. Peram, K. Veeramachaneni, C. K. Mohan, Fitness-distance-ratio based particle swarm optimization, in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, (2003), 174–181.
33. J. Kennedy, R. Mendes, Population structure and particle swarm performance, in *Proceedings of the 2002 Congress on Evolutionary Computation*, (2002), 1671–1676.
34. T. Hayashida, I. Nishizaki, S. Sekizaki, Y. Takamori, Improvement of Particle Swarm Optimization Focusing on Diversity of the Particle Swarm, in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, (2020), 191–197.
35. X. Xia, L. Gui, G. He, B. Wei, Y. Zhang, F. Yu, et al., An expanded particle swarm optimization based on multi-exemplar and forgetting ability, *Inf. Sci.*, **508** (2020), 105–120. https://doi.org/10.1016/j.ins.2019.08.065

36. X. Jin, Y. Liang, D. Tian, F. Zhuang, Particle swarm optimization using dimension selection methods, *Appl. Math. Comput.*, **219** (2013), 5185–5197. https://doi.org/10.1016/j.amc.2012.11.020

37. Y. Wang, W. Dong, X. Dong, A novel ITO Algorithm for influence maximization in the large-scale social networks, *Future Gener. Comput. Syst.*, **88** (2018), 755–763. https://doi.org/10.1016/j.future.2018.04.026

38. M. Ghasemi, E. Akbari, A. Rahimnejad, S. E. Razavi, S. Ghavidel, L. Li, Phasor particle swarm optimization: a simple and efficient variant of PSO, *Soft Comput.*, **23** (2019), 9701–9718.

AIMS Press