Information Sciences 540 (2020) 175–201

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Heterogeneous comprehensive learning and dynamic multi-swarm particle swarm optimizer with two mutation operators

Shengliang Wang [a,b], Genyou Liu [a,*], Ming Gao [a,b], Shilong Cao [a,b], Aizhi Guo [a], Jiachen Wang [a,b]

[a] State Key Laboratory of Geodesy and Earth's Dynamics, Institute of Geodesy and Geophysics, Chinese Academy of Sciences, Wuhan 430077, China
[b] College of Earth and Planetary Sciences, University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Article history:
Received 7 January 2020
Received in revised form 6 June 2020
Accepted 10 June 2020
Available online 25 June 2020

Keywords:
Particle swarm optimization (PSO)
Comprehensive learning (CL)
Dynamic multi-swarm (DMS)
Mutation operator
Exploitation
Exploration

ABSTRACT

In this paper, a heterogeneous comprehensive learning and dynamic multi-swarm particle swarm optimizer with two mutation operators (HCLDMS-PSO) is presented. In addition, a comprehensive learning (CL) strategy with the global optimal experience of the whole population is conducted to generate an exploitation subpopulation exemplar. However, a modified dynamic multi-swarm (DMS) strategy is specially designed to construct the exploration subpopulation exemplar. In the canonical DMS strategy, it is unfavorable for different sub-swarms to use the same linear decreasing inertia weight parameter. We first propose classifying the DMS sub-swarms at the search level and then constructing a novel nonlinear adaptive decreasing inertia weight for different sub-swarms, introducing a non-uniform mutation operator to enhance its exploration capability. Finally, the *gbest* of the whole population also adopts a Gaussian mutation operator to avoid falling into the local optimum. The particles of the two subpopulations will update their velocity independently without crippling one another to prevent a loss of diversity. The performance of HCLDMS-PSO is compared with those of 8 other PSO variants and 11 evolutionary algorithms on two classical benchmark optimization problems and a real-world engineering problem. Experimental results demonstrate that the HCLDMS-PSO improves the convergence speed, accuracy, and reliability on most optimization problems.

1. Introduction

Particle swarm optimization (PSO) is a swarm intelligence (SI) algorithm proposed by Kennedy and Eberhart in 1995 [32]. PSO was inspired by simulating bird flocking and fish schooling, and is a population-based search stochastic optimization method used to find the optimal solution to a single-objective problem. Owing to its simplicity, fast convergence speed, and few adjustable parameters, PSO has attracted the attention of numerous researchers in different areas during the past two decades. PSO has been successfully applied to many engineering fields, e.g., task assignment [11], image processing [35], and resource allocation [10], etc.

PSO suffers from diversity loss and premature convergence [39], and thus its performance is unsatisfying when the optimization problem has a large number of local optima or has a large number of dimensions and is non-separable. To improve the canonical PSO optimization performance, many scholars have proposed various PSO variants. The revision strategies for

* Corresponding author at: State Key Laboratory of Geodesy and Earth's Dynamics, Institute of Geodesy and Geophysics, Chinese Academy of Sciences, 340 Xudong St., Wuhan, PR China.
E-mail address: liugy@whigg.ac.cn (G. Liu).

https://doi.org/10.1016/j.ins.2020.06.027
0020-0255/

PSO can be roughly categorized into four cases: parameter tuning, neighborhood topology, learning strategy, and a hybridization of PSO with other algorithms.

(1) Parameter tuning

Shi and Eberhart [36] introduced an inertia weightω, which decreases linearly from a relatively large value to a small value over the optimization process to achieve a balance between global exploration and local exploitation. In addition, Clerc et al. [8] proposed the use of constriction coefficients to ensure the convergence of the PSO algorithm, and Ratnaweera [33] proposed a self-organizing hierarchical particle swarm optimizer (HPSO-TVAC) adopting the time-varying acceleration coefficients $c_1$and c_2. In FST-PSO [28], a novel fuzzy logic (FL) approach is used to calculate the inertia weight, cognitive/social factor, and minimum/maximum velocity independently for each particle, thus realizing a complete parameter settings-free. Using OLPSO [50], Zhan et al. identified four evolutionary statuses by evaluating the population distribution and particle fitness, namely, exploration, exploitation, convergence, and jumping out. This enables automatically controlling the inertia weight, acceleration coefficients, and other parameters to improve the search efficiency and convergence speed. In APSO-VI [48], Xu adjusted the inertia weight dynamically according to the average absolute velocity, which follows a given nonlinear ideal velocity through feedback control. In Ref. [40], various numerical experiments demonstrated that an adaptive self-regulating inertia weight algorithm can achieve a faster convergence and provide better solutions.

(2) Neighborhood topology

The particle neighborhood topology affects the information interaction between individuals, controls the exploration and exploitation, and drives the whole population evolution. The neighborhood topology can be divided into a static topology structure and a dynamic topology structure. In a static topology structure, the global best (*gbest)* and local best (*lbest)* are two classical models. Mendes and Kennedy [23] studied ring, four-cluster, pyramid and square structures, in which each individual is only connected with a few individuals within its neighborhood. The authors proposed a fully informed particle swarm (FIPS), in which each individual is affected by other individuals within the whole population.

A dynamic topology structure is more in line with the evolution characteristics of social groups and is more conducive to the information flow and interaction of the individuals to enhance the swarm diversity. Liang et al. [17] proposed a dynamic multi-swarm particle swarm optimizer (DMS-PSO), which divides the whole population into many small swarms, which are frequently regrouped using various regrouping schedules. Using DNLPSO [27], Nasir et al. selected an exemplar particle from the local neighbors, and thus the learner particle might be affected by the historical information of its neighborhood or sometimes that of its own information. Jie et al. [13] proposed a multi-swarm PSO that takes advantage of a mixed search behavior to maintain the swarm diversity and introduces a cooperative mechanism to prompt the information exchange among sub-swarms. Lim et al. [20] presented a new PSO with an increasing topology connectivity (PSO-ITC) to achieve better control of the exploration/exploitation searches. Moreover, Passaro et al. [29] proposed a clustering approach to achieve a niche PSO, namely, k-means-based PSO (kPSO), in which particles communicate with each other and realize a parallel search of the optimal solution based on the niche information.

(3) Learning strategy

Liang et al. [18] proposed a novel comprehensive learning particle swarm optimization (CLPSO), which enhances the diversity of the population and discourages premature convergence by encouraging each particle to learn from different particles on different dimensions. Other scholars have proposed different modification strategies based on CLPSO to balance the exploration and exploitation. Among them, Lynn et al. [22] proposed heterogeneous comprehensive learning particle swarm optimization (HCLPSO) based on CLPSO. HCLPSO divides a swarm into exploration subpopulation and exploitation subpopulations, both of which adopt a CL strategy. Li et al. [15] proposed a self-learning particle swarm optimization (SLPSO), which adopts four different learning strategies based on an adaptive learning mechanism. Wang et al. [42] proposed an enhanced PSO algorithm called GOPSO, which employs generalized opposition-based learning (GOBL) and a Cauchy mutation strategy to prevent premature convergence. In ATLPSO-ELS [19], Lim et al. proposed an elitist learning strategy and an orthogonal learning strategy to ensure that the particles escape the local optimum. In addition, Xu et al. [47] proposed a dimensional learning strategy (DLS) and a two-swarm learning (TSL) strategy to avoid the two "oscillation" and "two steps forward, one step back" phenomena, thereby significantly improving the search efficiency. Cheng et al. [7] proposed social learning particle swarm optimization (SL-PSO), which performs well on low-dimensional problems and is promising for solving large-scale problems. In addition, Wang et al. [43] proposed a novel surrogate-assisted (SA) particle swarm optimization inspired through committee-based active learning (CAL).

(4) Hybridization of PSO with other algorithms

The SI optimization algorithm has become an extremely active field of research with favorable characteristics from different optimization algorithms combined to enhance the performance of PSO. The hybridization of PSO with other algorithms can be divided into two categories: a combination with other evolutionary algorithms and the introduction of specific search operators. For the First, PSO combined with a differential evolution (DE) [34] and an artificial bee colony (ABC) [16] not only maintain its diversity, it also accelerate the convergence speed, which can effectively balance the exploration and exploitation. For the second, specific search operators are introduced into PSO, including mutation operators [47], two differential mutations [5], chaos-based initialization and robust update mechanisms [41], and some nature-inspired operators, such as aging mechanism [4].

In the PSO algorithm, previous studies revealed that a proper balance must be struck between exploration and exploitation when finding the optimal solution to a problem [22]. Therefore, an over-emphasis on exploration may cause the population to waste time in the suboptimal region of the search space and slow down the convergence rate. On the opposite side,

an over-emphasis on the exploitation may lead to a significant loss of diversity and allow the algorithm to fall into the local optimum during the early stage of the search process. Many novel strategies have been designed to balance exploration and exploitation [21,22,47]. Stimulated by such methods, unlike the HCLPSO algorithm, only a CL strategy is adopted to construct a heterogeneous exploration subpopulation and exploitation subpopulation. Dynamic or variable structures not only make a population more robust and adaptive, they also make a population more conducive to an information flow and an interaction allowing the individual to enhance the swarm diversity. Adopting different strategies to balance the exploration and exploitation, instead of relying on a single strategy, may reveal the strengths and potential of each individual approach. Therefore, we take full advantage of the strengths of the CLPSO and DMS-PSO algorithms to balance the exploration and exploitation. A heterogeneous comprehensive learning and dynamic multi-swarm particle swarm optimizer with two mutation operators, called HCLDMS-PSO, is proposed in this paper.

With the proposed HCLDMS-PSO algorithm, the CLPSO algorithm with the *gbest* of the whole population is adopted to mainly generate an exploitation subpopulation exemplar. Their particles learn from their different historical experiences on different dimensions and simultaneously learn from the *gbest* of the whole population. The DMS-PSO algorithm is assigned to generate an exploration subpopulation, in which particles conduct a local version PSO and sub-swarms frequently regroup to exchange information in a timely manner. To improve the DMS subpopulation exploration capability during the early evolutionary stage and enhance the exploitation capability in the latter evolutionary stage, a nonlinear adaptive decreasing inertia weight is more suitable than a linear decreasing inertia weight (LDIW) [45]. Therefore, a nonlinear inertia weight formula based sigmoid function is proposed. The overall state of different DMS sub-swarms should be evaluated simultaneously and a suitable inertia weight can be determined adaptively to match the current state of the sub-swarm. A non-uniform mutation operator is widely used in the GA [6]. We also introduced a non-uniform mutation operator in a DMS subpopulation, which mutates the position of particles with a lesser probability *Pm* in certain dimensions to increase the diversity of a subpopulation. This can effectively help the DMS subpopulation particles jump out of the local optimum. Finally, the *gbest* of the whole population introduced another mutation operator called a Gaussian mutation operator [47] to prevent from becoming trapped in the local optimum.

Two classical benchmark test suite functions with a total of 54 global optimization problems taken from CEC2005 and CEC2017, and real-world cases of a wireless sensor networks coverage control problem, are used in a performance evaluation of the proposed HCLDMS-PSO algorithm. The experiment results show that the proposed versions of the PSO algorithm achieve a proper balance between exploration and exploitation, and clearly improve the convergence accuracy, speed, and reliability. The remainder of this paper is organized as follows. Section 2.1 reviews the canonical PSO and provides the formulae for the global and local PSO. The CLPSO (CL strategy) and DMS-PSO (DMS strategy) are briefly introduced in Sections 2.2 and 2.3, respectively. The proposed HCLDMS-PSO algorithm and its computational complexity are presented in detail in Sections 3.1 and 3.2. Section 4 describes the extensive experiments conducted between the HCLDMS-PSO and other existing state-of-the-art evolutionary algorithms on two benchmark test suites and a real-world application problem. Finally, some concluding remarks are presented in Section 5.

2. Related studies

2.1. Canonical PSO

In canonical PSO, each particle is regarded as an individual solution to a single-objective problem without mass or volume in the D-dimensional search space and has its own position and velocity. First, the population particles are randomly initialized in terms of the position and velocity throughout the search space. Then, the fitness function is used to continuously aggregate each particle toward the best position in its own historical best experience *pbest* and *gbest* of the whole population. The formula of the particle swarm evolution is as follows:

$$v_i^d(t+1) = \omega * v_i^d(t) + c_1 * r_1 * \left[pbest_i^d(t) - x_i^d(t)\right] + c_2 * r_2 * \left[gbest_i^d(t) - x_i^d(t)\right] \tag{1}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{2}$$

where $d = 1, 2, ..., D$ is the dimensions of the parameters to be optimized. The ith particle position and velocity in the population ($i = 1, 2, ..., N$) are $x_i^d(t)$ and $v_i^d(t)$, respectively. In addition, $pbest_i = (pbest_i^1, pbest_i^2, \cdots, pbest_i^D)$ is the best historical position for the ith particle and $gbest = (gbest^1, gbest^2, \cdots, gbest^D)$ is the best position of the whole population. Here, $\omega$is an inertia weight parameter used to achieve a balance between the global exploration and local exploitation, which decreases linearly from a large value to a small value. Moreover, c_1 and c_2 are the acceleration coefficients, and r_1 and r_2 are two random numbers generated within the range of [0, 1]. In this way, the particle velocity update is composed of three parts. The first part is the inertia as the particle maintains its previous velocity, which indicates the "memory". The second part is the cognition of the particles themselves. The third part is the information sharing and cooperation between particles, which is expressed as "socialization".

The above formula is for the global PSO, and another formula is used for the local PSO. For the local PSO, each particle adjusts its velocity according to its historical best position, *pbest*, and the neighbor's best position, *lbest*. Its velocity update formula is modified as follows:

$$v_i^d(t+1) = \omega * v_i^d(t) + c_1 * r_1 * \left[pbest_i^d(t) - x_i^d(t)\right] + c_2 * r_2 * \left[lbest_i^d(t) - x_i^d(t)\right] \quad (3)$$

2.2. CLPSO

In the canonical PSO, particles learn from their optimal position *pbest* and global optimal position *gbest*. If *gbest* does not reach the optimal solution, all particles will easily fall into the local optimum and tend to convergence prematurely. In the CLPSO algorithm, all particles update their velocity using *pbest* and learn from the *pbest* of different particles in different dimensions [18], which is also called a CL strategy. The velocity of the *i*th particle is updated by Eq. (4):

$$v_i^d(t+1) = \omega * v_i^d(t) + c_1 * r_1 * \left[pbest_{fi(d)}^d(t) - x_i^d(t)\right] \quad (4)$$

where $f_i(d) = [f_i(1), f_i(2), \cdots f_i(D)]$ defines which particle's *pbest* the *i*th particle should follow for the *d*th dimension. Different particles define different learning probability values *Pc*. The *Pc* value of each particle is calculated through Eq. (5):

$$Pc_i = a + b * \frac{(\exp(10(i-1)/(ps-1)) - 1)}{(\exp(10) - 1)} \quad (5)$$

where "*ps*" represents the population size, $a = 0.05$, and $b = 0.45$. Each dimension of a particle generates a random number between [0, 1] and compares it with *Pc*. If the random number is greater than *Pc*, the particle learns the corresponding dimension of its *pbest*. If the random number is smaller than *Pc*, the particle will learn from the *pbest* of the corresponding dimension of the other particle. The guidance particle is determined by a tournament selection, i.e., two particles are selected and the particle with better fitness of the corresponding dimensions is chosen. Therefore, the exemplar $pbest_{fi(d)}^d$ is a new position that guides the particles toward the new direction in the search space. To ensure that the function evaluations are not wasted in the incorrect direction, a certain number of evaluations are defined as the refresh gap *m*. If the consecutive moves in generation *m* are not improved, a new $pbest_{fi(d)}^d$ will be generated until the particle ceases to improve. In this way, the best historical information of all particles is utilized to update the flying direction of the particle. This ensures that the CLPSO algorithm can preserve the population diversity and prevent a premature convergence.

2.3. DMS-PSO

DMS-PSO is a novel PSO algorithm with a dynamic topological structure [17], which divides the whole population into many small sub-swarms and the local version of PSO is used to update velocity and position of the particles. A better position is searched for using the members. These small sub-swarms are dynamic and regrouped, and their information is exchanged in a timely manner using a regrouping schedule.

In the DMS-PSO algorithm, each small sub-swarm uses its members to search for a better area within the space, which has a unique advantage to slow down the population's convergence velocity and increase the diversity. For every R generation, the population is regrouped at random and starts searching in another space using a new configuration of small sub-swarms. Here, R is called the regrouping period, which is a key parameter having a significant influence on the optimization results. Thus, R should neither be too small nor too large. The former aims to avoid an insufficient number of iterations for each sub-swarm to complete an adequate search. The latter aims to avoid function evaluations from being wasted if the sub-swarms cannot further improve. In this way, the information obtained by each small sub-swarm is exchanged among the whole population. Simultaneously, the diversity of the population will also increase. If not, the neighborhood structures of the sub-swarms will not change, and there will be no information exchange among the sub-swarms, resulting in a co-evolutionary PSO with the sub-swarms searching in parallel. The whole population can only search the local optimum and fall into a premature convergence, and it cannot search the global optimal solution.

The DMS-PSO algorithm is described in detail in Fig. 1. Suppose a population size of N = 9 and randomly divide it into three sub-swarms (n = 3). Then, each particle operates within the sub-swarm to search for a better solution. Its position is updated through Eqs. (2) and (3). However, the sub-swarms may converge to a local optimum in the regrouping schedule R. The whole population is then regrouped into three different sub-swarms, and the particles start updating again, with the process then repeated. Therefore, each small sub-swarm search space is enlarged, and the new small sub-swarms can find a better solution; thus, the DMS-PSO might perform better on complex problems. Details of the flow of the DMS-PSO algorithm can be found in the original study [17].

3. Proposed method

3.1. Proposed HCLDMS-PSO algorithm

To improve the solution performance of the PSO algorithm of unimodal or multimodal problems, the modified CL and DMS updating strategies are jointly used to update the position information of the particles described in this paper. Many existing evolutionary algorithms, such as the GA and ant colony algorithm, require larger populations, whereas

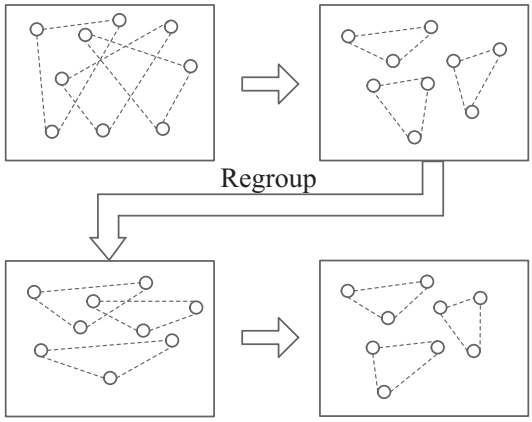


Fig. 1. DMS-PSO algorithm regrouping phase.

PSO only requires a relatively small population [17,18,33]. In general, a swarm of three to five particles can obtain satisfactory results for simple problems. Thus, the HCLDMS-PSO algorithm divides the whole population into two heterogeneous subpopulations for exploitation and exploration, respectively. The first subpopulation exemplar employs a CL strategy guided by the *gbest* of the whole population to enhance the exploitation capability. This is consistent with the exploitation subpopulation used by the HCLPSO algorithm. The second subpopulation assigns the DMS strategy modified by a nonlinear adaptive inertia weight, and the non-uniform mutation operator updates the particles position to enhance the exploration capability. Finally, the *gbest* of the whole population introduces a Gaussian mutation operator as a local optimization operator to enhance the local exploration capability of the HCLDMS-PSO algorithm.

In the HCLDMS-PSO algorithm, an exemplar of the exploitation subpopulation is generated by adopting CL strategies with the *gbest* of the whole population. Its velocity is updated using Eq. (6):

$$v_i^d(t+1) = \omega_1(t) * v_i^d(t) + c_1 * r_1 * \left[pbest^d_{fi(D)}(t) - x_i^d(t)\right] + c_2 * r_2 * \left[gbest^d(t) - x_i^d(t)\right] \tag{6}$$

where the inertia weight $\omega_1(t)$ is used as a linearly decreasing function of the number of iterations from 0.99 to 0.2, and the time-varying acceleration coefficients $c_1 = 2.5 \sim 0.5$ and $c_2 = 0.5 \sim 2.5$. Here, $gbest^d$ is the global optimal position of the whole population in the corresponding dimensions. By combining the CL strategy with *gbest* of the whole population, the CL subpopulation loses its population diversity but encourages the particles to quickly converge to the global optimum. Thus, the subpopulation has a strong exploitation capability.

The second subpopulation of the HCLDMS-PSO algorithm employs the DMS strategy described in Section 2.3 above, which is a local PSO with a new neighborhood topology. Thus, the DMS subpopulation achieves a significant improvement in terms of the global exploration but lacks a local exploitation capability. Although the DMS-PSO algorithm can efficiently solve multimodal problems, certain drawbacks still exist. The particles information among the different sub-swarms cannot be exchanged until the population is regrouped, and the sub-swarms only learn from their *pbest* and local best experience *lbest*. During the process of optimization, the capabilities of the different small sub-swarm particles in a DMS subpopulation are generally different, which creates the different sub-swarms at different search levels. Therefore, it is inappropriate for different sub-swarms to use the same linear decreasing inertia weightω. Aiming at this drawback, we propose classifying the search level of all sub-swarms and constructing a novel nonlinear adaptive inertia weight ω to improve the particle exploration and exploitation capability of the DMS subpopulation. For example, for a sub-swarm with a poor search performance, it is necessary to eliminate the improper position as quickly as possible and accelerate the exploration of the other areas. For a sub-swarm with a better search performance, it is necessary to strengthen the exploitation capability of the local areas and excavate better solutions. The inertial weight ω plays an extremely important role in individual optimization, and thus we adjust this parameter adaptively according to the overall state of the different dynamic sub-swarms to improve their particle optimization capability. To better describe the process of this modified strategy, we first provide a parameter definition.

Definition1:. Recording the sub-swarm i asSub_i, N_i denotes the size of Sub_i. Note that $f_{i,j}$ denotes the fitness of the jth particle inSub_i. Then, the average fitness value of Sub_i is defined as$m_i = \sum_{j=1}^{N_i} f_{i,j}/N_i$. Note that f_j denotes the fitness of the jth particle within the whole population. The average fitness value of the whole population is then defined as$M = \sum_{j=1}^{N} f_j/N$.

The sigmoid function shows an excellent balance between linearity and nonlinearity, which is a favorable threshold function. The sigmoid function is often used to construct the neuron activation function in a neural network and is widely applied in machine learning and artificial intelligence [12], and is defined as Eq. (7).

$$S(t) = \frac{1}{1+\exp(-t)} \tag{7}$$

The sigmoid function is continuous, smooth, and strictly monotone, and its range is limited to (0, 1). When t exceeds the range of [−10, 10], the value of function $S(t)$ remains unchanged. Thus, we first construct a nonlinear decreasing inertia weight (NLDIW) ω calculation formula based on the sigmoid function, as follows:

$$w_1(t) = w_{\max} + (w_{\min} - w_{\max})/1 + \exp\left[-5 \times \left(\frac{2t}{T} - 1\right)\right] \tag{8}$$

where $w_{\max} = 0.99$ and $w_{\min} = 0.20$ are the maximum and minimum inertia weights, respectively, t is the current run time, and T is the maximum evolution time. We then adjust the proposed adaptive nonlinear decreasing inertia weight (ANLDIW) based on the total number of search levels (Definition 1) of different sub-swarms in the DMS subpopulation. This can be divided into the following two cases:

$$\begin{cases} \omega_2(t) = \omega_1(t) + C, m_i \geqslant M \\ \omega_2(t) = \omega_1(t) - C, m_i < M \end{cases} \tag{9}$$

In Eq. (9), for the minimizing problem, if $m_i \geqslant M$, the sub-swarm is in a poor position and far from the optimal solution. Therefore, their particles should improve the ability of the global exploration and avoid a premature convergence. This is realized by appropriately increasing the nonlinear decreasing inertial weight $\omega_1(t)$ by a constant value C, and if $\omega_2(t)$ exceeds the maximum inertia weight $w_{\max} = 0.99$, by taking $\omega_2(t) = w_{\max}$. If $m_i < M$, the sub-swarm is in a better position and close to the optimal solution. Thus, their particles should slow down their velocity and avoid missing the optimal solution with an overly large search step. We adjust the step size by appropriately reducing the nonlinear decreasing inertial weight $\omega_1(t)$ by a constant value C, and if $\omega_2(t)$ exceeds the minimum inertia weight $w_{\min} = 0.20$, by taking $\omega_2(t) = w_{\min}$. The value of parameter C is discussed in detail in Section 4.1.

We also introduce a non-uniform mutation operator in the DMS subpopulation to avoid a premature convergence and an increase in the DMS subpopulation diversity. The non-uniform mutation operator is used to search for tiny areas near an individual particle with a small mutation probability Pm = 0.1, which is defined through Eqs. (10) and (11).

$$x' = x + \mathbf{N}_{d\times d}(\mathbf{B_U} - x)\left(1 - \frac{t}{T}\right)^b, \textit{if } \mathbf{round}(rand) = 0 \tag{10}$$

$$x' = x + \mathbf{N}_{d\times d}(x - \mathbf{B_L})\left(1 - \frac{t}{T}\right)^b, \textit{if } \mathbf{round}(rand) = 1 \tag{11}$$

where x and x' are an individual particle before and after the non-uniform mutation operator is applied, t is the current run time, T is the maximum evolution time, d is the particle dimension, and $\mathbf{N}_{d\times d}$ is a diagonal matrix whose elements have a uniform distribution of between zero and 1. In addition, $\mathbf{B_U}$ and $\mathbf{B_L}$are the upper and lower boundaries of the particle search space, respectively, and b is a system parameter dependent on the evolutionary times, and generally takes a value of 2~5. Here, "**round**" indicates a rounding and "rand" is a uniformly distributed random number independently generated within [0, 1]. The detailed process of the non-uniform mutation operator is shown in Algorithm 1. According to Eqs. (10) and (11) and Algorithm 1, the non-uniform mutation conducts a uniform random search at the initial stage of evolution for the DMS subpopulation and upon a local disturbance at the later stage of the evolution. Thus, the non-uniform mutation operator can effectively improve the diversity of the DMS subpopulation and enhance the global exploration and local exploitation.

In addition, to prevent the whole population from falling into the local optima, we introduce a Gaussian mutation operator on the *gbest* of the whole population. Together with the above-described characteristics and process, a pseudocode of the HCLDMS-PSO is presented in detail in Algorithm 2. The MATLAB code of this algorithm is available on the GitHub platform (https://github.com/wangshengliang2018).

Algorithm 1. *Non-uniform mutation operator*

```
for each particle i do
  for each dimension j do
    if rand < Pm then
      a = rand(1, D); N_{d×d} = diag(a);
      if round (rand) = 0 then
        Update the particle position according Eq. (10).
      else round (rand) = 1 then
        Update the particle position according Eq. (11).
      end if
    end if
  end for
end for
```

Algorithm 2. *HCLDMS-PSO phase*

```
Set parameter N, N1, N2, n;
Initialize X_i and V_i (1 ≤ i ≤ N);
Evaluate X_i and record the fitness fit(X_i), fes = N;
Divide the DMS subpopulation into m sub-swarms randomly, with n particles in each swarm;
pbest_i = X_i; gbest = [pbest_i |min(fit(pbest_i)),1 ≤ i ≤ N];
lbest_k = [pbest_i |min(fit(pbest_i)),1 ≤ i ≤ n, (1 ≤ k ≤ m)];
While (fes <= MaxFEs)
  for i = 1:N1
    Update the CL subpopulation X_i and V_i according to Eqs. (6) and (2);
  end for
  Calculate M = mean(fit(X_i)), m_i = mean(fit(X_{i,j})); Calculate ω_2(t) according Eqs. (8) and (9);
  for i = 1:N2
    Update the DMS subpopulation X_i and V_i according to Eqs. (3) and (2);
    X_i perform the non-uniform mutation by Eqs. (10) and (11);
  end for
  X = [X_1, X_2] and V = [V_1, V_2] ;
  Evaluate all X_i fitness;
  Update the pbest_i and gbest;
  for i = 1:m
    Update the DMS subpopulation sub-swarms lbest_i;
  end for
  gbest perform the Gaussian mutation operator;
  Updating the exemplar pbest^d_{fi(d)}(t) for the CL subpopulation according to Section 2.2;
  Regrouping the DMS subpopulation sub-swarms particles according to Section 2.3;
End While
```

The population diversity can accurately evaluate the ability of the exploitation and exploration. To further illustrate the effectiveness of the exploitation and exploration capabilities of two subpopulations in the proposed HCLDMS-PSO algorithm, we analyze the diversity of the CL subpopulation, the DMS subpopulation, and the whole population during the evolution process. The calculation formula of the diversity measure is described in Eqs. (12) and (13).

$$Diversity = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\sum_{d=1}^{D}\left(X_i^d(t) - \overline{X^d}\right)^2} \tag{12}$$

$$\overline{X^d} = \frac{1}{N}\sum_{i=1}^{N} X_i^d(t) \tag{13}$$

where N is the size of the population, D indicates the dimensions of the search space, X_i^d is the dth dimension of the ith particle, and $\overline{X^d}$ denotes the average position of the dth dimension of the population. A small diversity of the population indicates that the particles converge near the center of the population, that is, they are under an exploitation within a small space. A large diversity of the population indicates that the particles are far from the center of the population, that is, a larger space is explored.

As shown in Fig. 2, the unimodal (F1 and F5) and multimodal (F7 and F10) functions in the CEC2005 benchmark functions [37] are selected from the median indicator of 30 runs for the population diversity analysis. The search space dimensions, the population size, and the max fitness evaluations (FEs) are 30, 40, and 300,000, respectively. It can be clearly seen in the diversity graphs that the DMS subpopulation mostly has a higher diversity than the CL subpopulation, and that the whole population diversity ranks between the CL subpopulation and the DMS subpopulation. The CL subpopulation maintains the smaller diversity, and consequently converges rapidly, which is consistent with the exploitation subpopulation in the HCLPSO algorithm. In the F5 function, the CL subpopulation diversity may also be greater than the DMS subpopulation within a short time but rapidly decreases to lower than the DMS subpopulation during the later stage of the evolution. The DMS subpopulation maintains a higher diversity as expected because its particles are guided by the local optimum, and the nonlinear adaptive decreasing inertia weight and a non-uniform mutation operator are adopted. Finally, the whole population diversity ranks in the middle, owing to the balance of the exploitation and exploration provided by the cooperation of the CL and DMS subpopulations. Therefore, the experimental results used to compare the diversity satisfy our expectation that the CL subpopulation is mainly responsible for the exploitation, whereas the DMS subpopulation is mainly

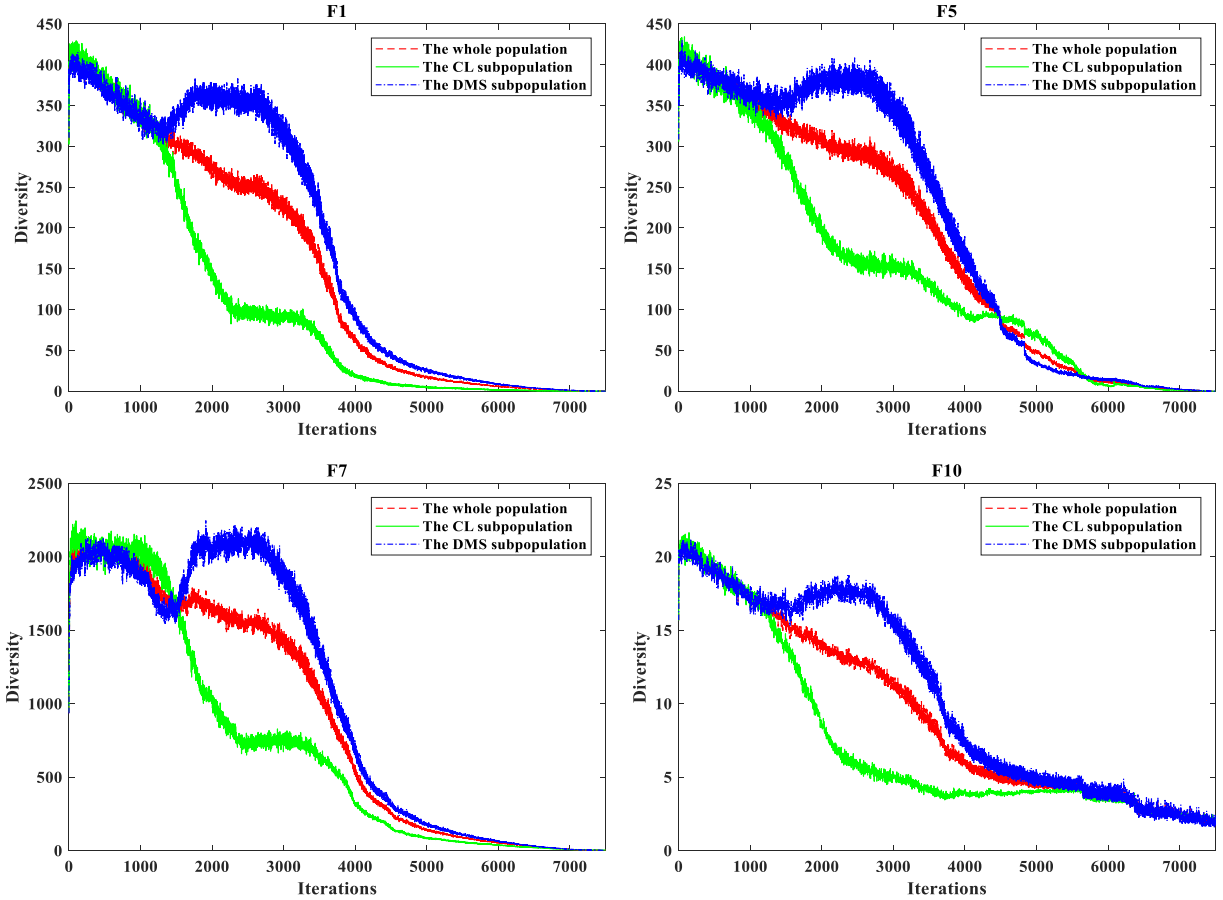


Fig. 2. Diversity measure value comparisons of the CL subpopulation, the DMS subpopulation, and the whole population.

responsible for the global exploration. The interaction and cooperation of the two subpopulations ensure a global and rapid convergence of the whole population.

3.2. Algorithmic computational complexity analysis

The computational costs of the canonical PSO algorithm include the initialization, fitness evaluation, and velocity and position updates, which have complexities of $O(mn)$, $O(mn)$, and $O(2mn)$, respectively (where m and n are the swarm size and dimensions, respectively). Thus, the time complexity of the canonical PSO is $O(mn)$. The HCLDMS-PSO algorithm computationally includes updating the CL subpopulation learning exemplars, and the DMS subpopulation executes a non-uniform mutation operator. The worst-case time complexity of the construction of the CL subpopulation learning exemplars is $O(m_1 n)$ (where m_1 is the population size of the CL subpopulation). In addition, the worst-case time complexity of the DMS subpopulation executing a non-uniform mutation operator is $O(m_2 n)$ (where m_2 is the population size of the DMS subpopulation). According to the above component complexity analysis, HCLDMS-PSO has the same time complexity $O(mn)$ as the canonical PSO algorithm.

4. Experimental results and discussions

4.1. Parameter tuning

In the proposed HCLDMS-PSO algorithm, we need to calibrate two subpopulation group sizes (*N1* and *N2*) and a DMS subpopulation with different sub-swarm sizes *n*. The second parameter that needs to be calibrated is the ANLDIW of the DMS subpopulation. The parameters are calibrated using 25 shifted and rotated CEC 2005 test suite functions [37]. The set of CEC2005 benchmark functions is comprised of 25 different types of unimodal, multimodal, expanded, and hybrid composition functions. The search range, initialization range, global optimum and bias values of all benchmark functions are listed in Table 1. The parameter tuning used the same run time of 30, population size of 20, FES of 100,000, and 10 dimensions. The results are ranked based on the mean error and standard deviation, and the average rank value of 25 benchmark functions is counted as the final rank.

Table 1
CEC2005 Test suite functions.

Function type	Test functions	Initialization range	Search range	x^*	$F(x^*)f\_bias$
Unimodal functions	F1: Shifted Sphere Function	$[-100,100]^D$	$[-100,100]^D$	o	−450
	F2: Shifted Schwefel's Problem 1.2	$[-100,100]^D$	$[-100,100]^D$	o	−450
	F3: Shifted Rotated High Conditioned Elliptic Function	$[-100,100]^D$	$[-100,100]^D$	o	−450
	F4: Shifted Schwefel's Pdroblem1.2 with Noise in Fitness	$[-100,100]^D$	$[-100,100]^D$	o	−450
	F5: Schwefel's Problem2.6 with Global Optimum on Bounds	$[-100,100]^D$	$[-100,100]^D$	o	−310
Multimodal functions	F6: Shifted Rosenbrock's Function	$[-100,100]^D$	$[-100,100]^D$	o	390
	F7: Shifted Rotated Griewank's Function without Bounds	$[0,600]^D$	$[-600,600]^D$	o	−180
	F8: Shifted Rotated Ackley's Function with Global Optimum on Bounds	$[-32,32]^D$	$[-32,32]^D$	o	−140
	F9: Shifted Rastrigin's Function	$[-5,5]^D$	$[-5,5]^D$	o	−330
	F10: Shifted Rotated Rastrigin's Function	$[-5,5]^D$	$[-5,5]^D$	o	−330
	F11: Shifted Rotated Weierstrass Function	$[-0.5,0.5]^D$	$[-0.5,0.5]^D$	o	90
	F12: Schwefel's Problem 2.13	$[-100,100]^D$	$[-100,100]^D$	o	−460
Expanded functions	F13: Expanded Extended Griewank's plus Rosenbrock's Function (F8F2)	$[-3,1]^D$	$[-3,1]^D$	o	−130
	F14: Shifted Rotated Expanded Scaffer's F6	$[-100,100]^D$	$[-100,100]^D$	o	−300
Hybrid composition functions	F15: Hybrid Composition Function	$[-5,5]^D$	$[-5,5]^D$	o	120
	F16: Rotated Hybrid Composition Function	$[-5,5]^D$	$[-5,5]^D$	o	120
	F17: Rotated Hybrid Composition Function with Noise in Fitness	$[-5,5]^D$	$[-5,5]^D$	o	120
	F18: Rotated Hybrid Composition Function	$[-5,5]^D$	$[-5,5]^D$	o	10
	F19: Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum	$[-5,5]^D$	$[-5,5]^D$	o	10
	F20: Rotated Hybrid Composition Function with the Global Optimum on the Bounds	$[-5,5]^D$	$[-5,5]^D$	o	10
	F21: Rotated Hybrid Composition Function	$[-5,5]^D$	$[-5,5]^D$	o	360
	F22: Rotated Hybrid Composition Function with High Condition Number Matrix	$[-5,5]^D$	$[-5,5]^D$	o	360
	F23: Non-Continuous Rotated Hybrid Composition Function	$[-5,5]^D$	$[-5,5]^D$	o	360
	F24: Rotated Hybrid Composition Function	$[-5,5]^D$	$[-5,5]^D$	o	260
	F25: Rotated Hybrid Composition Function without Bounds	$[-2,5]^D$	$[-5,5]^D$	o	260

$o = [o_1, o_2, o_3, \cdots, o_D]$: shifted global optimum; x^*: global optimum, D = dimension.

Table 2
Calibration of two subpopulation group sizes (*N1* and *N2*) and different sub-swarm size *n* of the DMS subpopulation.

Functions	Criteria	11 + 9 (n = 3)	8 + 12 (n = 3)	5 + 15 (n = 3)	12 + 8 (n = 4)	8 + 12 (n = 4)	4 + 16 (n = 4)	15 + 5 (n = 5)	10 + 10 (n = 5)	5 + 15 (n = 5)
F1	mean	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	std	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
F2	mean	3.22e−14	2.46e−14	2.84e−14	6.63e−14	3.60e−14	**1.33e−14**	2.69e−13	4.55e−14	2.27e−14
	std	3.23e−14	2.86e−14	2.89e−14	7.17e−14	3.16e−14	**2.45e−14**	2.39e−13	4.81e−14	2.83e−14
	rank	5	3	4	8	6	**1**	9	7	2
F3	mean	6.10e+04	6.84e+04	5.78e+04	7.63e+04	7.55e+04	7.92e+04	8.93e+04	**4.56e+04**	6.38e+04
	std	4.37e+04	4.54e+04	3.45e+04	5.40e+04	4.23e+04	4.86e+04	6.60e+04	**3.47e+04**	4.91e+04
	rank	3	5	2	7	6	8	9	**1**	4
F4	mean	1.05e−06	5.21e−11	5.48e−13	6.94e−05	1.66e−09	**2.27e−14**	1.05e−04	1.35e−05	2.46e−14
	std	3.61e−06	2.23e−10	9.53e−13	9.71e−05	8.75e−09	**3.20e−14**	3.36e−04	3.20e−05	2.86e−14
	rank	6	4	3	8	5	**1**	9	7	2
F5	mean	1.09e−08	2.91e−12	**1.58e−12**	2.90e−01	1.96e−11	5.15e−12	3.89e−01	5.19e−01	5.40e−12
	std	2.53e−08	3.90e−12	**2.23e−12**	6.75e−01	4.84e−11	7.92e−12	9.69e−01	1.29e+00	5.18e−12
	rank	6	2	**1**	7	5	3	8	9	4
F6	mean	5.02	2.36	3.00	4.66	**1.98**	3.04	4.06	4.30	3.22
	std	7.54	1.80	6.54	8.31	**1.34**	5.21	6.68	6.80	8.72
	rank	9	2	3	8	**1**	4	6	7	5
F7	mean	8.44e−02	7.63e−02	**5.60e−02**	8.32e−02	8.48e−02	7.18e−02	1.02e−01	9.59e−02	8.43e−02
	std	5.28e−02	3.60e−02	**2.89e−02**	4.44e−02	4.36e−02	3.82e−02	6.65e−02	4.71e−02	5.62e−02
	rank	6	3	**1**	4	7	2	9	8	5
F8	mean	20.17	20.17	20.18	20.20	20.19	20.17	20.20	**20.17**	20.19
	std	0.08	0.07	0.07	0.07	0.07	0.08	0.06	**0.06**	0.09
	rank	2	4	5	8	7	3	9	**1**	6
F9	mean	**2.59**	2.92	3.46	3.18	3.48	3.78	3.12	2.89	3.55
	std	**1.49**	1.41	1.84	2.18	1.90	1.66	1.45	1.37	1.27
	rank	**1**	3	6	5	7	9	4	2	8
F10	mean	5.60	**4.84**	4.95	7.63	6.40	5.54	9.49	9.02	7.30
	std	2.78	**1.90**	1.73	3.69	2.98	2.26	4.05	3.35	2.47
	rank	4	**1**	2	7	5	3	9	8	6
F11	mean	2.11	2.02	**1.46**	2.15	1.79	1.60	3.08	2.21	1.73
	std	1.13	1.15	**0.76**	0.90	0.89	0.90	0.94	1.19	0.99
	rank	6	5	**1**	7	4	2	9	8	3
F12	mean	66.94	31.98	62.41	**7.65**	20.95	56.58	48.75	19.14	61.69
	std	191.30	128.77	184.90	**8.90**	52.71	244.39	146.06	52.13	246.57
	rank	9	4	8	**1**	3	6	5	2	7
F13	mean	0.57	0.55	0.54	0.50	0.53	0.55	0.58	**0.49**	0.55
	std	0.14	0.12	0.10	0.15	0.14	0.18	0.14	**0.16**	0.11
	rank	8	7	4	2	3	6	9	**1**	5
F14	mean	2.26	2.19	2.10	2.38	2.20	**1.98**	2.51	2.39	2.28
	std	0.56	0.48	0.56	0.59	0.57	**0.60**	0.50	0.52	0.48
	rank	5	3	2	7	4	**1**	9	8	6
F15	mean	1.61e+02	1.38e+02	1.50e+02	**1.10e+02**	1.29e+02	1.56e+02	1.80e+02	1.39e+02	1.29e+02
	std	1.47e+02	1.33e+02	1.39e+02	**1.10e+02**	1.24e+02	1.36e+02	1.63e+02	1.47e+02	1.32e+02
	rank	8	4	6	**1**	2	7	9	5	3
F16	mean	**1.01e+02**	1.02e+02	1.01e+02	1.04e+02	1.03e+02	1.01e+02	1.15e+02	1.08e+02	1.05e+02
	std	**1.11e+01**	6.15e+00	5.19e+00	1.29e+01	8.31e+00	5.99e+00	9.65e+00	8.38e+00	6.74e+00
	rank	**1**	4	2	6	5	3	9	8	7
F17	mean	**1.00e+02**	1.04e+02	1.04e+02	1.05e+02	1.07e+02	1.03e+02	1.14e+02	1.10e+02	1.05e+02
	std	**1.55e+01**	7.42e+00	7.17e+00	2.23e+01	7.84e+00	8.40e+00	1.26e+01	9.30e+00	7.23e+00
	rank	**1**	3	4	5	7	2	9	8	6
F18	mean	**6.42e+02**	6.78e+02	6.82e+02	7.07e+02	7.21e+02	7.03e+02	7.05e+02	7.10e+02	7.17e+02
	std	**2.37e+02**	2.07e+02	2.08e+02	2.25e+02	1.92e+02	2.06e+02	2.12e+02	2.20e+02	2.11e+02
	rank	**1**	2	3	6	9	4	5	7	8
F19	mean	7.51e+02	7.12e+02	6.87e+02	**6.71e+02**	7.21e+02	7.13e+02	7.60e+02	7.70e+02	7.31e+02
	std	1.57e+02	2.11e+02	2.09e+02	**2.29e+02**	1.88e+02	2.00e+02	1.78e+02	1.66e+02	2.01e+02
	rank	7	3	2	**1**	5	4	8	9	6
F20	mean	6.98e+02	6.81e+02	7.20e+02	7.04e+02	7.18e+02	7.03e+02	6.89e+02	**6.20e+02**	7.34e+02
	std	2.11e+02	2.37e+02	1.83e+02	2.13e+02	1.92e+02	2.07e+02	2.48e+02	**2.55e+02**	1.83e+02
	rank	4	2	8	6	7	5	3	**1**	9
F21	mean	3.79e+02	4.17e+02	4.43e+02	4.37e+02	**3.67e+02**	4.46e+02	3.85e+02	3.78e+02	4.30e+02
	std	1.32e+02	1.23e+02	1.47e+02	1.75e+02	**9.59e+01**	1.86e+02	1.30e+02	1.27e+02	1.51e+02
	rank	3	5	8	7	**1**	9	4	2	6
F22	mean	7.48e+02	7.14e+02	7.33e+02	7.49e+02	7.32e+02	**6.98e+02**	7.19e+02	7.17e+02	7.43e+02
	std	4.89e+00	1.13e+02	8.27e+01	1.64e+01	8.19e+01	**1.35e+02**	1.15e+02	1.13e+02	8.03e+00
	rank	8	2	6	9	5	**1**	4	3	7
F23	mean	6.41e+02	6.43e+02	6.49e+02	6.59e+02	**6.35e+02**	6.44e+02	6.54e+02	6.57e+02	6.62e+02

Table 2 (*continued*)

Functions	Criteria	11 + 9 (n = 3)	8 + 12 (n = 3)	5 + 15 (n = 3)	12 + 8 (n = 4)	8 + 12 (n = 4)	4 + 16 (n = 4)	15 + 5 (n = 5)	10 + 10 (n = 5)	5 + 15 (n = 5)
	std	9.02e+01	1.01e+02	1.01e+02	9.80e+01	**8.26e+01**	1.32e+02	1.01e+02	1.16e+02	1.15e+02
	rank	2	3	5	8	**1**	4	6	7	9
F24	mean	**200**	**200**	**200**	**200**	**200**	**200**	210	200	**200**
	std	**0**	**0**	**0**	**0**	**0**	**0**	54.77	5.81e−13	**0**
	rank	**1**	**1**	**1**	**1**	**1**	**1**	3	2	**1**
F25	mean	210	**200**	**200**	220	210	210	220	**200**	**200**
	std	54.77	**0.00**	**0.00**	76.11	54.77	54.77	76.11	**0.00**	**0.00**
	rank	2	**1**	**1**	4	2	2	3	**1**	**1**
Average rank		4.36	**3.08**	3.56	5.36	4.36	3.68	6.72	4.92	5.08
Final rank		4	**1**	2	7	4	3	8	5	6
Best/2nd Best/ Worst		6/2/2	4/5/0	6/5/0	5/1/2	5/1/1	6/4/2	1/0/13	6/4/2	3/2/2

The results of tuning the two subpopulation group sizes (*N1* and *N2*) and the DMS sub-population with a different sub-swarm size n are shown in Table 2. According to the final rank, two subpopulation group sizes of *N1* = 8 and *N2* = 12(n = 3) obtained the best performance for all 25 benchmark functions compared to other combinational settings.

For the inertia weight parameters of the DMS subpopulation, we compare the LDIW and the proposed ANLDIW for different constant values of $C = 0 \sim 0.2$. The results are shown in Table 3. It can be clearly seen that the LDIW achieves the worst performance compared with other ANLDIW in a DMS subpopulation. With ANLDIW, $C = 0.15$ provides the best performance for all 25 test functions according to final ranking.

Therefore, two subpopulation group sizes (*N1* = 8, *N2* = 12(n = 3)) and the DMS subpopulation ANLDIW $C = 0.15$ are selected for a further performance evaluation of the proposed HCLDMS-PSO algorithm.

4.2. Comparison with other PSO variants on CEC2005 test suite

The performance of the proposed HCLDMS-PSO algorithm is compared with those of eight other state-of-the-art PSO variants, namely, the inertia weight PSO, FDR-PSO, HPSO-TVAC, DMS-PSO, CLPSO, LIPS, HCLPSO, and EPSO.

The inertia weight PSO algorithm [36], which was introduced in detail in Section 2.1, only uses the inertia weight to balance the exploration and exploitation capabilities in finding the global optimum. FDR-PSO [30] adds social learning from the experience of the neighboring particles, called *nbest*, which has a better fitness. HPSO-TVAC [33] introduces time-varying acceleration coefficients to enhance the global exploration during the early stage and encourage convergence toward the global optimum during the latter stages of the optimization. Under this strategy, the previous velocity $v_i^d(t)$in Eq. (1) remains at zero, and the particles are reinitialized with random velocities whenever they stagnate in the search space. The CLPSO [18] and DMS-PSO [17] algorithms are introduced in Sections 2.2 and 2.3. The former uses the CL strategy and learning probability *Pc* curve, which encourages a particle to learn from different particles with different levels of exploration and exploitation. The latter achieves a balance between exploration and exploitation through a recombination of multiple subpopulations. The LIPS [31] algorithm adopts the local best information from its nearest neighborhood (measured based on the Euclidean distance) to guide the particles in searching for the global optimum. All particles are well distributed over the entire space during the exploration stage and a search is conducted within each Euclidean distance-based neighborhood without any interference from another niche during the exploitation stage. HCLPSO [22] is a CLPSO variant in which the whole population is divided into two subpopulations aiming to focus on exploration (only CL strategy) and exploitation-enhanced (CL strategy with the *gbest* of the whole population), respectively. The proposed HCLDMS-PSO algorithm constructs an exploitation subpopulation similar with HCLPSO, but the exploration subpopulation is conducted using the modified DMS strategy. EPSO [21] employs a self-adaptive scheme to ensemble five PSO strategies including the inertia weight PSO, CLPSO-*gbest*, FDR-PSO, HPSO-TVAC, and LIPS. Different PSO strategies are conducted through a self-adaptive selection based on their previous success and failure in finding a good solution.

The detailed parameter settings of all PSO variants mentioned above as well as the proposed HCLDMS-PSO are listed in Table 4.

To compare the proposed HCLDMS-PSO algorithm with other state-of-the-art PSO algorithms, the evaluation criteria are as follows: first, the performance of the mean error and standard deviation of each algorithm solution for 25 benchmark functions are calculated and ranked. The average rank of the 25 benchmark functions is counted as the final rank of each algorithm. The non-parametric Wilcoxon signed-rank test is used to examine the statistical difference between the HCLDMS-PSO algorithm and the other PSO variants. A Wilcoxon signed-rank test is a nonparametric hypothesis test used to compare two vector data with continuous symmetries about the median, and to test the hypothesis of the zero median

Table 3
Calibration of inertial weight parameter of the DMS subpopulation.

Functions	Criteria	LDIW	ANLDIW (C = 0)	ANLDIW (C = 0.05)	ANLDIW (C = 0.1)	ANLDIW (C = 0.15)	ANLDIW (C = 0.2)
F1	mean	**0**	**0**	**0**	**0**	**0**	**0**
	std	**0**	**0**	**0**	**0**	**0**	**0**
	rank	**1**	**1**	**1**	**1**	**1**	**1**
F2	mean	4.35e−14	2.46e−14	3.97e−14	3.03e−14	**1.32e−14**	3.22e−14
	std	3.55e−14	3.23e−14	5.99e−14	4.41e−14	**2.44e−14**	3.85e−14
	rank	6	2	5	3	**1**	4
F3	mean	7.25e+04	5.88e+04	4.64e+04	**4.57e+04**	5.65e+04	7.19e+04
	std	6.35e+04	4.40e+04	4.64e+04	**2.65e+04**	4.49e+04	6.21e+04
	rank	6	4	2	**1**	3	5
F4	mean	2.90e−08	4.55e−07	1.05e−07	3.09e−07	4.23e−12	**3.18e−12**
	std	1.09e−07	2.21e−06	5.57e−07	1.65e−06	1.96e−11	**1.20e−11**
	rank	3	6	4	5	2	**1**
F5	mean	1.64e−03	2.35e−02	2.41e−06	4.36e−09	2.60e−11	**1.78e−11**
	std	8.98e−03	1.29e−01	1.28e−05	2.20e−08	5.80e−11	**4.42e−11**
	rank	5	6	4	3	2	**1**
F6	mean	3.80	7.93	5.13	**2.01**	2.73	5.67
	std	6.14	17.84	9.15	**1.56**	3.23	16.07
	rank	3	6	4	**1**	2	5
F7	mean	9.37e−02	7.09e−02	**6.38e−02**	9.73e−02	8.99e−02	9.03e−02
	std	4.12e−02	3.64e−02	**2.80e−02**	3.34e−02	3.94e−02	4.20e−02
	rank	5	2	**1**	6	3	4
F8	mean	20.20	20.21	20.18	20.19	**20.17**	20.17
	std	8.67e−02	0.10	6.51e−02	8.94e−02	**6.56e−02**	6.65e−02
	rank	5	6	3	4	**1**	2
F9	mean	3.48	**2.78**	2.91	3.38	2.91	3.05
	std	1.70	**1.34**	1.58	1.32	1.42	1.77
	rank	6	**1**	3	5	2	4
F10	mean	7.76	**6.63**	7.06	7.03	7.16	7.39
	std	3.35	**2.42**	2.46	2.49	2.47	2.79
	rank	6	**1**	3	2	4	5
F11	mean	1.47	**1.43**	1.75	1.61	1.91	1.87
	std	1.12	**1.07**	1.18	0.85	1.10	0.91
	rank	2	**1**	4	3	6	5
F12	mean	**27.81**	30.75	33.52	31.15	89.85	1.12e+02
	std	**1.29e+02**	1.28e+02	1.28e+02	1.01e+02	3.09e+02	3.86e+02
	rank	**1**	2	4	3	5	6
F13	mean	0.54	0.56	0.52	**0.51**	0.57	0.59
	std	0.15	0.15	0.16	**0.17**	0.15	0.15
	rank	3	4	2	**1**	5	6
F14	mean	2.14	2.14	**2.03**	2.26	2.13	2.13
	std	0.41	0.59	**0.61**	0.41	0.62	0.52
	rank	4	5	**1**	6	3	2
F15	mean	1.62e+02	1.40e+02	1.33e+02	1.44e+02	1.38e+02	**1.27e+02**
	std	1.45e+02	1.48e+02	1.31e+02	1.49e+02	1.33e+02	**1.19e+02**
	rank	6	4	2	5	3	**1**
F16	mean	**97.32**	1.01e+02	1.03e+02	1.04e++02	1.04e+02	1.03e+02
	std	**22.48**	8.32	7.37	7.08	9.80	12.23
	rank	**1**	2	3	6	5	4
F17	mean	1.03e+02	1.05e+02	1.05e+02	**1.03e+02**	1.04e+02	1.07e+02
	std	7.50	10.39	7.70	**10.36**	5.56	8.32
	rank	2	5	4	**1**	3	6
F18	mean	7.23e+02	7.06e+02	6.87e+02	6.78e+02	**6.64e+02**	7.59e+02
	std	1.94e+02	2.09e+02	2.07e+02	2.24e+02	**2.28e+02**	1.76e+02
	rank	5	4	3	2	**1**	6
F19	mean	7.02e+02	6.70e+02	**6.22e+02**	7.10e+02	7.06e+02	6.98e+02
	std	2.00e+02	2.28e+02	**2.50e+02**	2.04e+02	2.09e+02	2.08e+02
	rank	4	2	**1**	6	5	3
F20	mean	6.63e+02	6.62e+02	6.87e+02	6.59e+02	**6.41e+02**	6.66e+02
	std	2.25e+02	2.30e+02	2.05e+02	2.25e+02	**2.39e+02**	2.35e+02
	rank	4	3	6	2	**1**	5
F21	mean	4.13e+02	**3.79e+02**	4.18e+02	3.94e+02	3.90e+02	3.86e+02
	std	1.00e+02	**1.31e+02**	1.28e+02	1.47e+02	1.24e+02	1.00e+02
	rank	5	**1**	6	4	3	2
F22	mean	7.45e+02	7.30e+02	7.43e+02	**6.99e+02**	7.42e+02	7.32e+02
	std	8.65	81.46	13.27	**1.35e+02**	10.58	81.99
	rank	6	2	5	**1**	4	3
F23	mean	6.30e+02	6.40e+02	**6.02e+02**	6.32e+02	6.41e+02	6.54e+02
	std	1.20e+02	82.25	**72.75**	1.01e+02	1.06e+02	1.13e+02

Table 3 (*continued*)

Functions	Criteria	LDIW	ANLDIW (C = 0)	ANLDIW (C = 0.05)	ANLDIW (C = 0.1)	ANLDIW (C = 0.15)	ANLDIW (C = 0.2)
	rank	2	4	**1**	3	5	6
F24	mean	**200**	**200**	**200**	**200**	**200**	**200**
	std	**0**	**0**	**0**	**0**	**0**	**0**
	rank	**1**	**1**	**1**	**1**	**1**	**1**
F25	mean	210	**200**	**200**	**200**	**200**	**200**
	std	54.77	**0**	**0**	**0**	**0**	**0**
	rank	2	**1**	**1**	**1**	**1**	**1**
Average rank		3.76	3.04	2.96	3.04	**2.88**	3.56
Final rank		5	3	2	3	**1**	4
Best/2nd Best/Worst		4/4/6	7/6/4	7/3/2	8/3/4	**7/4/1**	6/3/5

Table 4
Parameter settings of the nine PSOs.

Algorithm	Parameters setting	Year	Ref.
PSO	w:0.9 ~ 0.4, c1 = c2 = 2, Vmax = 0.5*Range	1995	[36]
FDR-PSO	w:0.9 ~ 0.4, c1 = c2 = 1, c = 2, Vmax = 0.2*Range	2003	[30]
HPSO-TVAC	c1 = 2.5 ~ 0.5, c2 = 0.5 ~ 2.5, Vmax = 0.5*Range	2004	[33]
DMS-PSO	w:0.729, c1 = c2 = 1.49445, Vmax = 0.5*Range	2005	[17]
CLPSO	w:0.9 ~ 0.2, c1 = c2 = 1.49445, Vmax = 0.2*Range	2006	[18]
LIPS	w:0.729, Vmax = 0.5*Range, nsize = 3	2012	[31]
HCLPSO	w:0.99 ~ 0.29,c1:2.5 ~ 0.5,c2:0.5 ~ 2.5,K:3 ~ 1.5,Vmax = 0.5*Range	2015	[22]
EPSO	–	2017	[21]
HCLDMS-PSO	w:0.99 ~ 0.29,w_2 ,c1 = 2.5 ~ 0.5, c2 = 0.5 ~ 2.5, Pm = 0.1,Vmax = 0.5*Range	–	–

for the difference between paired samples [9]. Here, the HCLDMS-PSO algorithm and other PSO variants are conducted using a Wilcoxon signed-rank test at a 5% significance level. In the single objective optimization problem, a pairwise comparison is conducted over the solution results obtained through 30 simulation runs on 10- and 30-dimensional problems. The results are shown in Tables 6 and 8, respectively. The symbol "+" indicates that the HCLDMS-PSO algorithm performs significantly better than the compared PSO algorithms. The symbol "0" indicates that the HCLDMS-PSO algorithm is not significantly different from the compared algorithm. The symbol "−" indicates that the compared algorithm performs significantly better than the HCLDMS-PSO algorithm. The numbers of the signs "+/0/−" are counted for each algorithm and are presented in the last rows in Tables 6 and 8.

Finally, the convergence process of all PSO algorithms for unimodal, multimodal, expanded and hybrid composition functions is analyzed. The median indicator of 30 runs for the 30-dimensional problems is used to analyze the convergence performance. As shown in Figs. 3 and 4, the convergence characteristic graphs of the HCLDMS-PSO algorithm and other state-of-the-art PSO algorithms for some CEC2005 benchmark functions are presented.

4.2.1. Results for 10 dimensional problems

Experiments on the 10-dimensional problems were conducted using 100,000 function evaluations (FES) and a population size of 20. Subpopulation sizes of 8/12 are utilized for the CL and DMS subpopulations, respectively. Other parameter settings are as described in Table 4 and Section 4.1. The experimental results are illustrated in Table 5.

As shown in Table 5, for unimodal problems, all PSO algorithms have the same result on function F1 except for the last three algorithms. DMS-PSO achieve the best solution on F2, F4, and F5. FDR achieves the best solution on function F3. DMS-PSO and FDR achieve the best performances for the unimodal function overall, whereas the proposed HCLDMS-PSO performs the second-best overall and is ranked second on functions F2 and F3. For multimodal, expanded, and hybrid composition functions, the HCLDMS-PSO algorithm ranks first on functions F7, F10-F12, F14, F16, F17, F20, F24, and F25, i.e., 10 out of the 20 test functions. The HCLPSO algorithm also achieves the best performance on functions F9, F15, F18, F19, F21, and F22, i.e., 6 out of the 20 functions. The CLPSO algorithm achieves the best results on functions F6, F9, F13, and F23, whereas the DMS algorithm obtains the best performance on function F8. The number of best, second best, and worst scores are counted for each algorithm and presented in the last row of Table 5. It can be clearly seen that the HCLDMS-PSO, EPSO, HCLPSO, and DMS-PSO algorithms avoid receiving the worst performance on any of the benchmark functions. The proposed HCLDMS-PSO algorithm offers the best performance on a total of 11 out of the 25 benchmark problems, although its average rank was only slightly higher than that of the HCLPSO algorithm, i.e., with a final rank of second overall on 10-dimensional problems. The foremost reason for this result is the unsatisfactory performance provided by the HCLDMS-PSO algorithm in comparison with the HCLPSO algorithm on test functions F6, F9, F13, F18, and F23 for such problems. The Wilcoxon signed rank test results of the HCLDMS-PSO algorithm for 10-dimensional problems compared with the other algorithms are shown

Table 5
Comparison of experimental of PSO algorithm for 10 dimensional CEC2005 test functions.

Functions	Criteria	HCLDMS-PSO	EPSO	HCLPSO	LIPS	CLPSO	DMS-PSO	HPSO-TVAC	FDR	PSO
F1	mean	**0**	**0**	**0**	**0**	**0**	**0**	2.19e−13	2.27e−14	3.41e−14
	std	**0**	**0**	**0**	**0**	**0**	**0**	1.56e−13	2.83e−14	2.83e−14
	rank	**1**	**1**	**1**	**1**	**1**	**1**	4	2	3
F2	mean	2.84e−14	8.14e−14	5.15e−13	1.47e−07	4.2962e−04	**0**	1.16e−12	7.95e−14	1.00e−13
	std	2.89e−14	5.31e−14	3.06e−13	8.06e−07	1.00e−03	**0**	5.72e−13	5.50e−14	4.13e−14
	rank	2	4	6	8	9	**1**	7	3	5
F3	mean	5.54e+04	5.75e+04	6.50e+04	2.74e+06	2.83e+05	1.66e+05	7.29e+04	**4.29e+04**	1.45e+05
	std	3.82e+04	4.25e+04	4.80e+04	6.47e+06	1.31e+05	1.08e+05	6.39e+04	**3.50e+04**	1.27e+05
	rank	2	3	4	8	9	7	5	**1**	6
F4	mean	4.01e−08	2.54e−07	5.95e−04	1.25e+03	0.32	**0**	7.53e+02	2.69e−11	1.61e−13
	std	2.19e−07	1.01e−06	1.68e−03	1.27e+03	0.65	**0**	9.19e+02	1.31e−10	1.06e−13
	rank	4	5	6	9	7	**1**	8	3	2
F5	mean	8.87e−10	6.42e−04	1.65	2.69e+03	1.55	**1.57e−12**	6.50e+02	6.15e−10	1.25e+03
	std	4.64e−09	1.70e−03	2.63	8.44e+02	2.97	**1.95e−12**	4.07e+02	1.04e−09	2.13e+02
	rank	3	4	6	9	5	**1**	7	2	8
F6	mean	2.04	0.73	1.26	62.20	**0.56**	2.70	15.20	1.52	10.91
	std	1.42	1.63	2.33	176.98	**0.72**	4.54	40.05	3.63	15.56
	rank	5	2	3	9	**1**	6	8	4	7
F7	mean	**7.04e−02**	0.18	0.10	4.49	0.17	8.66e−02	2.69	0.26	0.25
	std	**3.39e−02**	8.68e−02	4.78e−02	4.73	9.43e−02	5.13e−02	1.79	0.23	0.21
	rank	**1**	5	3	9	4	2	8	7	6
F8	mean	20.18	20.31	20.23	20.29	20.32	**20.15**	20.16	20.27	20.31
	std	0.06	0.09	0.07	0.12	0.07	**0.04**	0.09	0.09	0.09
	rank	3	7	4	6	8	**1**	2	5	7
F9	mean	3.28	**0**	**0**	9.25	**0**	0.23	0.36	2.75	1.95
	std	1.41	**0**	**0**	3.07	**0**	0.50	0.55	1.32	1.09
	rank	6	**1**	**1**	7	**1**	2	3	5	4
F10	mean	**7.89**	8.45	8.75	10.08	8.29	8.40	34.79	10.84	13.73
	std	**2.89**	2.22	2.94	4.13	2.16	2.13	13.92	4.37	5.40
	rank	**1**	4	5	6	2	3	9	7	8
F11	mean	**2.01**	3.25	2.62	3.84	4.23	4.82	6.15	2.39	3.33
	std	**1.12**	1.40	1.17	1.31	0.68	0.92	1.13	1.37	1.24
	rank	**1**	4	3	6	7	8	9	2	5
F12	mean	**11.26**	20.30	13.84	2.25e+03	48.96	92.34	3.59e+02	1.50e+03	1.46e+03
	std	**45.65**	33.34	34.94	5.65e+03	45.11	242.98	6.56e+02	2.54e+03	3.20e+03
	rank	**1**	3	2	9	4	5	6	8	7
F13	mean	0.56	0.48	0.32	0.65	**0.32**	0.51	0.58	0.67	0.55
	std	0.13	0.14	0.08	0.20	**0.05**	0.10	0.25	0.21	0.27
	rank	6	3	2	8	**1**	4	7	9	5
F14	mean	**2.09**	2.59	2.62	3.33	3.08	2.89	3.12	2.76	2.96
	std	**0.57**	0.54	0.40	0.36	0.23	0.31	0.34	0.59	0.51
	rank	**1**	2	3	9	7	5	8	4	6
F15	mean	1.15e++02	9.52	**5.61**	2.91e+02	12.09	1.15e+02	2.17e+02	2.96 e+02	2.96e+02
	std	1.25e+02	44.84	**21.76**	1.10e+02	25.18	1.23e+02	2.13e+02	1.71 e+02	1.89e+02
	rank	5	2	**1**	7	3	4	6	8	9
F16	mean	**1.04e+02**	1.10e+02	1.08e+02	1.26e+02	1.13e+02	1.12e+02	1.94e+02	1.72e+02	1.49e+02
	std	**10.37**	9.01	8.05	17.45	9.21	9.86	44.73	1.79e+02	86.38
	rank	**1**	3	2	6	5	4	9	8	7
F17	mean	**1.04 e+02**	1.12e+02	1.13e+02	1.38e+02	1.26e+02	1.17e+02	2.19e+02	1.45e+02	1.41e+02
	std	**9.38**	14.27	14.36	26.15	10.64	11.58	71.83	1.06e+02	62.33
	rank	**1**	2	3	6	5	4	9	8	7
F18	mean	7.21e+02	6.31e+02	**5.04e+02**	9.04e+02	6.26e+02	7.45e+02	9.98e+02	8.47e+02	7.00e+02
	std	1.93e+02	2.24e+02	**2.27e+02**	1.14e+02	1.88e+02	2.10e+02	91.77	1.65e+02	2.55e+02
	rank	5	3	**1**	8	2	6	9	7	4
F19	mean	7.07e+02	6.34e+02	**6.00e+02**	9.24e+02	6.05e+02	7.62e+02	9.20e+02	8.07e+02	7.13e+02
	std	1.93e+02	2.01e+02	**2.25e+02**	1.03e+02	1.85e+02	1.99e+02	1.40e+02	2.15e+02	2.48e+02
	rank	4	3	**1**	9	2	6	8	7	5
F20	mean	**6.30e+02**	6.68e+02	6.38e+02	8.99e+02	6.91e+02	7.70e+02	9.93e+02	7.87e+02	7.83e+02
	std	**2.40e+02**	1.96e+02	2.11e+02	1.50e+02	1.61e+02	1.95e+02	84.80	2.47e+02	1.94e+02
	rank	**1**	3	2	8	4	5	9	7	6
F21	mean	4.30e+02	4.24e+02	**3.84e+02**	7.48e+02	4.05e+02	6.36e+02	1.00e+03	8.06e+02	5.06e+02
	std	1.51e+02	1.95e+02	**1.02e+02**	2.77e+02	98.86	2.35e+02	3.15e+02	3.30e+02	2.19e+02
	rank	4	3	**1**	7	2	6	9	8	5
F22	mean	7.29e+02	7.25e+02	**6.95e+02**	8.08e+02	7.34e+02	7.49e+02	8.88e+02	7.72e+02	7.80e+02
	std	81.27	1.16e+02	**1.57e+02**	50.73	1.18e+02	86.96	76.91	1.03e+02	97.72
	rank	3	2	**1**	8	4	5	9	6	7
F23	mean	6.79e+02	6.21e+02	5.92e+02	8.92e+02	**5.62e+02**	7.94e+02	1.11e+03	9.30e+02	6.43e+02
	std	1.36e+02	1.23e+02	91.48	2.70e+02	**91.69**	2.05e+02	2.18e+02	2.38e+02	1.02e+02

Table 5 (*continued*)

Functions	Criteria	HCLDMS-PSO	EPSO	HCLPSO	LIPS	CLPSO	DMS-PSO	HPSO-TVAC	FDR	PSO
	rank	5	3	2	7	**1**	6	9	8	4
F24	mean	**2.00e+02**	2.00e+02	2.00e+02	3.88e+02	2.00e+02	2.10e+02	7.24e+02	3.73e+02	2.83e+02
	std	**0**	1.39e−12	1.11e−12	3.07e+02	1.70e−05	54.77	4.36e+02	1.96e+02	1.68e+02
	rank	**1**	3	2	8	4	5	9	7	6
F25	mean	**2.00e+02**	2.00e+02	2.00e+02	4.32e+02	2.00e+02	2.30e+02	9.81e+02	3.50e+02	2.10e+02
	std	**0**	8.07e−13	1.11e−12	3.63e+02	3.20e−10	91.53	4.29e+02	2.19e+02	54.77
	rank	**1**	4	2	9	3	5	7	8	6
Average rank		2.72	3.16	**2.68**	7.48	4.04	4.12	7.36	5.76	5.8
Final rank		2	3	**1**	9	4	5	8	6	7
Best/2nd Best/Worst		**11/2/0**	2/5/0	**7/7/0**	1/0/9	5/4/3	5/2/0	0/1/11	1/3/1	0/1/1

Table 6
Wilcoxon signed rank test results of single-problem analysis with a significance level of $\alpha = 0.05$ for 10 dimensional problems.

Functions	Pairwise comparison HCLDMS-PSO versus							
	EPSO	HCLPSO	LIPS	CLPSO	DMS-PSO	HPSO-TVAC	FDR	PSO
F1	0	0	0	0	0	+	+	+
F2	+	+	+	+	−	+	+	+
F3	0	0	+	+	+	0	0	+
F4	+	+	+	+	−	+	0	0
F5	+	+	+	+	−	+	−	+
F6	−	−	+	−	0	+	−	+
F7	+	+	+	+	0	+	+	+
F8	+	+	+	+	0	0	+	+
F9	−	−	+	−	−	−	0	−
F10	0	0	+	0	0	+	+	+
F11	+	0	+	+	+	+	0	+
F12	+	+	+	+	+	+	+	+
F13	−	−	0	−	0	0	0	0
F14	+	+	+	+	+	+	+	+
F15	−	−	+	−	0	0	+	+
F16	0	0	+	+	+	+	+	+
F17	0	+	+	+	+	+	+	+
F18	0	−	−	0	+	+	+	0
F19	0	0	+	0	0	+	+	0
F20	0	0	+	0	+	+	+	+
F21	0	0	+	0	+	+	+	0
F22	−	−	+	+	+	+	+	+
F23	0	−	+	−	0	+	+	0
F24	0	0	+	+	0	+	+	+
F25	0	0	+	0	0	+	+	+
+/0/−	8/12/5	8/10/7	22/2/1	13/7/5	10/11/4	20/4/1	18/5/2	18/6/1

in Table 6. The HCLDMS-PSO algorithm performs better than the other PSO algorithms on most of the benchmark test functions, except for a few functions such as F6, F9, and F13.

4.2.2. Results for 30-dimensional problems

An experiment on 30-dimensional problems was conducted using an FES of 300,000 and a population size of 40. Subpopulation sizes of 16/24 were used for the CL subpopulation and DMS subpopulation, respectively. The other parameter settings also were as described in Section 4.1 and Table 4. The experiment results are illustrated in Table 7.

As shown in Table 7, for 30-dimensional unimodal problems, all PSO algorithms perform well on function F1. The proposed HCLDMS-PSO achieves the best solution on F4. EPSO achieves the best solution on F2 and F3. FDR achieves the best solution on function F5. EPSO and FDR respectively perform the best and second-best on the unimodal function in terms of their overall performance, whereas the proposed HCLDMS-PSO performs the third-best overall and ranks second on function F5.

For multimodal problems, the HCLDMS-PSO algorithm ranks first on functions F7, F10, and F11 and second on functions F8 and F12. For another, CLPSO yields the best solution on F6 and F9, whereas DMS-PSO achieves the best solution on function F8. HPSO-TVAC offers the best solution for F12, and overall, HCLDMS-PSO obtains the best overall performance on the multimodal problems. For the expanded and hybrid composition functions, HCLDMS-PSO achieved the best results on 9 out of 13 test functions, i.e., F14, F16-F19 and F22-F25. In addition, HCLPSO achieves the best results on F13 and F20, whereas CLPSO performs the best on F15, and EPSO offers the best results on F21. Hence, HCLDMS-PSO consistently performs well

Table 7
Comparison of experimental of PSO algorithm for 30 dimensional CEC2005 test functions.

Functions	Criteria	HCLDMS-PSO	EPSO	HCLPSO	LIPS	CLPSO	DMS-PSO	HPSO-TVAC	FDR	PSO
F1	mean	5.87e−14	5.49e−14	8.33e−14	1.44e−13	5.49e−14	**1.89e−15**	8.67e−13	1.15e−13	1.45e−13
	std	1.03e−14	1.03e−14	2.88e−14	1.47e−13	1.03e−14	**1.03e−14**	5.48e−13	3.49e−14	4.13e−14
	rank	3	2	4	6	2	**1**	8	5	7
F2	mean	3.00e−02	**8.90e−13**	8.22e−05	4.31	1.09e+03	21.44	3.67e−06	2.25e−11	0.43
	std	3.56e−02	**2.87e−13**	1.09e−04	10.16	2.81e+02	13.80	3.00e−06	6.23e−11	0.88
	rank	5	**1**	4	7	9	8	3	2	6
F3	mean	1.13e+06	**3.23e+05**	9.39e+05	1.24e+07	1.27e+07	5.89e+06	8.44e+05	3.75e+05	7.71e+06
	std	3.95e+05	**1.41e+05**	5.40e+05	1.52e+07	4.05e+06	2.59e+06	3.18e+05	1.77e+05	5.64e+06
	rank	5	**1**	4	8	9	6	3	2	7
F4	mean	**43.79**	1.09e+02	6.68e+02	3.20e+04	8.57e+03	1.35e+03	2.45e+04	3.08e+02	4.44e+02
	std	**45.38**	1.14e+02	3.41e+02	6.68e+03	1.89e+03	5.39e+02	5.49e+03	1.72e+02	3.19e+02
	rank	**1**	2	5	9	7	6	8	3	4
F5	mean	2.29e+03	4.26e+03	3.20e+03	8.51e+03	4.40e+03	2.41e+03	8.55e+03	**3.31e+02**	4.35e+03
	std	3.70e+02	8.30e+02	6.14e+02	1.70e+03	3.96e+02	4.09e+02	1.65e+03	**5.58e+02**	7.02e+02
	rank	2	5	4	8	7	3	9	**1**	6
F6	mean	26.64	1.34	2.46	1.24e+02	**1.04**	52.25	64.02	9.28	64.49
	std	21.56	1.57	2.03	2.37e+02	**1.64**	38.22	52.28	22.72	86.51
	rank	5	2	3	9	**1**	6	7	4	8
F7	mean	**6.66e−03**	2.20e−02	1.94e−02	95.42	0.73	1.75e−02	1.10e−02	1.53e−02	0.66
	std	**9.80e−03**	2.12e−02	1.48e−02	41.19	0.10	1.06e−02	6.28e−03	1.25e−02	0.55
	rank	**1**	6	5	9	8	4	2	3	7
F8	mean	20.68	20.89	20.81	20.89	20.89	**20.53**	20.72	20.86	20.91
	std	0.12	8.25e−02	6.67e−02	0.18	7.07e−02	**4.65e−02**	1.07e−01	7.80e−02	7.06e−02
	rank	2	7	4	8	6	**1**	3	5	9
F9	mean	29.71	0.33	1.09e−13	52.93	**5.68e−14**	13.73	10.88	30.91	17.04
	std	5.28	0.65	3.31e−14	13.09	**0**	10.19	5.50	7.89	4.16
	rank	7	3	2	9	**1**	5	4	8	6
F10	mean	**28.55**	57.70	56.09	67.86	1.03e+02	54.05	2.84e+02	58.39	84.95
	std	**5.99**	14.99	14.21	23.06	16.48	6.59	51.58	14.20	51.95
	rank	**1**	4	3	6	8	2	9	5	7
F11	mean	**10.34**	23.04	21.04	20.73	24.83	28.28	29.54	18.48	21.92
	std	**3.54**	3.38	3.01	2.58	1.30	1.49	2.47	3.96	3.56
	rank	**1**	6	4	3	7	8	9	2	5
F12	mean	3.97e+03	4.89e+03	4.80e+03	6.62e+04	1.33e+04	6.78e+03	**3.69e+03**	7.05e+03	4.45e+04
	std	3.70e+03	4.46e+03	4.54e+03	4.23e+04	4.08e+03	5.49e+03	**3.99e+03**	8.85e+03	3.76e+04
	rank	2	4	3	9	7	5	**1**	6	8
F13	mean	2.59	2.21	**1.47**	3.54	1.82	3.95	3.02	3.01	2.96
	std	0.52	0.40	**0.22**	0.82	0.19	0.51	0.55	0.67	0.71
	rank	4	3	**1**	8	2	9	7	6	5
F14	mean	**11.16**	11.83	11.72	12.43	12.64	12.44	12.30	11.67	12.63
	std	**0.64**	0.61	0.75	0.36	0.20	0.24	0.51	0.77	0.30
	rank	**1**	4	3	6	9	7	5	2	8
F15	mean	3.23e+02	1.08e+02	89.29	3.95e+02	**37.46**	3.56e+02	3.57e+02	3.13e+02	3.33e+02
	std	78.42	97.06	87.76	96.60	**20.40**	81.70	1.51e+02	1.10e+02	96.29
	rank	5	3	2	9	**1**	7	8	4	6
F16	mean	**63.18**	1.15e+02	1.05e+02	1.56e+02	1.57e+02	2.62e+02	3.78e+02	2.46e+02	2.22e+02
	std	**36.36**	51.85	37.88	65.68	22.23	1.90e+02	1.26e+02	1.81e+02	1.44e+02
	rank	**1**	3	2	4	5	8	9	7	6
F17	mean	**1.23e+02**	1.33e+02	1.28e+02	2.86e+02	2.31e+02	1.78e+02	4.20e+02	2.95e+02	3.43e+02
	std	**1.38e+02**	76.34	49.22	99.31	33.91	1.26e+02	1.11e+02	1.98e+02	1.28e+02
	rank	**1**	3	2	6	5	4	9	7	8
F18	mean	**8.91e+02**	9.01e+02	8.96e+02	9.69e+02	9.10e+02	9.09e+02	9.03e+02	9.16e+02	9.27e+02
	std	**41.82**	46.39	43.79	17.36	20.57	1.73	79.59	4.08	3.30
	rank	**1**	3	2	9	6	5	4	7	8
F19	mean	**8.88e+02**	9.00e+02	9.10e+02	9.67e+02	9.15e+02	9.08e+02	9.21e+02	9.16e+02	9.26e+02
	std	**44.98**	45.60	20.95	37.19	1.32	2.04	74.15	4.33	1.86
	rank	**1**	2	4	9	5	3	7	6	8
F20	mean	9.06e+02	9.11e+02	**8.98e+02**	9.68e+02	9.12e+02	9.08e+02	9.12e+02	9.16e+02	9.27e+02
	std	20.21	30.62	**39.46**	15.81	15.15	2.43	72.16	3.60	2.46
	rank	2	4	**1**	9	5	3	6	7	8
F21	mean	5.00e+02	**4.97e+02**	4.99e+02	5.21e+02	5.00e+02	5.00e+02	1.11e+03	5.81e+02	5.10e+02
	std	1.85e−13	**14.30**	3.60	1.16e+02	3.50e−13	2.49e−13	2.83e+02	1.62e+02	54.77
	rank	3	**1**	2	7	5	4	9	8	6
F22	mean	**9.05e+02**	9.22e+02	9.06e+02	1.02e+03	9.73e+02	9.09e+02	1.21e+03	9.26e+02	9.44e+02
	std	**10.41**	20.75	16.39	42.93	15.35	12.08	66.52	16.80	14.18
	rank	**1**	4	2	8	7	3	9	5	6
F23	mean	**5.34e+02**	5.34e+02	5.34e+02	6.42e+02	5.34e+02	5.34e+02	1.19e+03	6.69e+02	5.60e+02
	std	**1.89e−04**	4.16e−04	4.01e−04	1.67e+02	3.15e−04	2.37e−04	1.91e+02	2.17e+02	1.01e+02

Table 7 (*continued*)

Functions	Criteria	HCLDMS-PSO	EPSO	HCLPSO	LIPS	CLPSO	DMS-PSO	HPSO-TVAC	FDR	PSO
	rank	**1**	5	4	7	3	2	9	8	6
F24	mean	**2.00e+02**	2.00e+02	2.00e+02	2.81e+02	2.00e+02	2.00e+02	1.30e+03	2.00e+02	2.00e+02
	std	**1.32e−13**	1.46e−13	1.62e−12	2.58e+02	3.07e−12	7.93e−13	26.03	1.43e−12	1.43e−12
	rank	**1**	2	5	7	6	3	8	4	4
F25	mean	**2.00e+02**	2.00e+02	2.00e+02	4.59e+02	2.00e+02	2.00e+02	1.31e+03	2.00e+02	2.10e+02
	std	**2.95e−13**	5.86e−13	1.56e−12	4.03e + 02	4.45e−10	5.72e−13	33.61	1.10e−12	54.77
	rank	**1**	3	5	8	6	2	9	4	7
Average rank		**2.32**	3.32	3.2	7.52	5.48	4.6	6.6	4.84	6.64
Final rank		**1**	3	2	9	6	4	7	5	8
Best/2nd Best/Worst		**13/4/0**	3/5/0	2/7/0	0/0/9	3/2/3	2/3/1	1/1/11	1/4/0	0/0/1

Table 8
Wilcoxon signed rank test results of single-problem analysis with a significance level of $\alpha = 0.05$ for 30 dimensional problems.

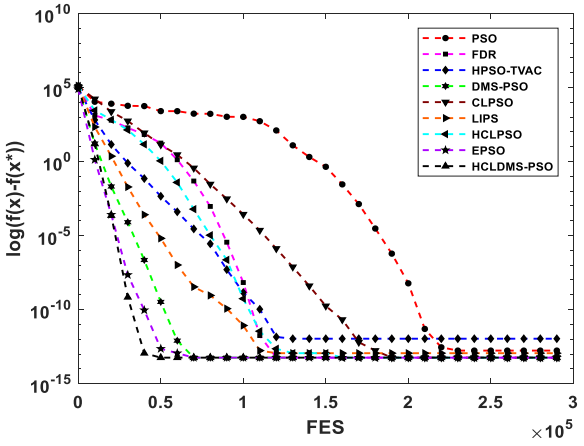
Functions	Pairwise comparison HCLDMS-PSO versus							
	EPSO	HCLPSO	LIPS	CLPSO	DMS-PSO	HPSO-TVAC	FDR	PSO
F1	0	+	+	0	−	+	+	+
F2	−	−	+	+	+	−	−	+
F3	−	−	+	+	+	−	−	+
F4	+	+	+	+	+	+	+	+
F5	+	+	+	+	0	+	−	+
F6	−	−	+	−	+	+	−	0
F7	+	+	+	+	+	+	+	+
F8	+	+	+	+	−	0	+	+
F9	−	−	+	−	−	−	0	−
F10	+	+	+	+	+	+	+	+
F11	+	+	+	+	+	+	+	+
F12	0	0	+	+	+	0	0	+
F13	−	−	+	−	+	+	+	+
F14	+	+	+	+	+	+	+	+
F15	−	−	+	−	0	0	0	0
F16	+	+	+	+	+	+	+	+
F17	0	0	+	+	+	+	+	+
F18	+	+	+	+	0	0	+	+
F19	+	+	+	+	0	0	+	+
F20	+	−	+	+	0	0	+	+
F21	−	−	+	+	+	+	+	+
F22	+	0	+	+	0	+	+	+
F23	+	0	+	0	+	+	+	+
F24	0	+	+	0	+	+	+	+
F25	0	+	+	+	+	+	+	+
+/0/−	13/5/7	13/4/8	25/0/0	18/3/4	16/6/3	16/6/3	18/3/4	22/2/1

throughout the expanded and hybrid composition functions and achieves the best overall performance. In total, depending on the final rank and the number of best, second best, and worst scores, the proposed HCLDMS-PSO algorithm achieves the highest performance, ranking first on 13 out of the 25 benchmark problems. The last row in the Wilcoxon sign rank test in Table 8 also shows that the proposed HCLDMS-PSO performs better than the other PSO algorithms on most of the 30-dimensional problems.

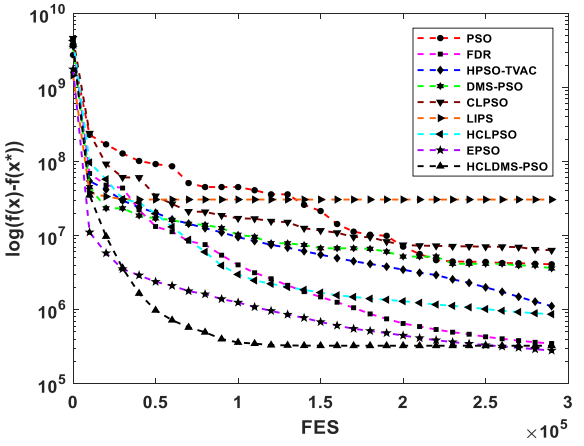
Furthermore, the characteristic graphs showing the convergence of the proposed HCLDMS-PSO algorithm are presented in Figs. 3 and 4. For unimodal and multimodal functions, we can see from Fig. 3 that the proposed HCLDMS-PSO achieves a higher convergence accuracy and faster convergence speed than the other PSO algorithms. The proposed HCLDMS-PSO algorithm can jump out of the local minimum and avoid a premature convergence, achieving either the best or second-best performance. Among the hybrid composite functions, the convergence graphs of F16, F17, F19, F22, F23, and F25 are presented in Fig. 4. The proposed HCLDMS-PSO algorithm offers an outstanding performance in F16, F17, and F19 and achieves a performance comparable to that of the other algorithms in F22, F23, and F25. To summarize, the convergence performance analysis demonstrates that the proposed HCLDMS-PSO algorithm outperforms the other PSO algorithms in most of the test cases.

4.3. Experimental results and analysis of CEC2017 test suite

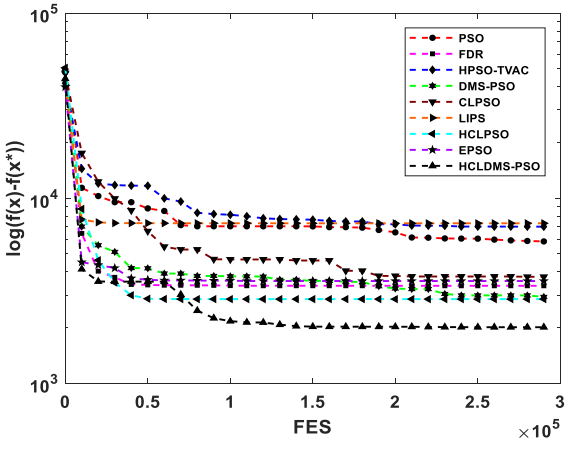
In this section, we also compare the performance of the proposed HCLDMS-PSO algorithm with other state-of-the-art evolutionary algorithms in Table 9 (including the bat algorithm (BA) [49], grey wolf optimizer (GWO) [25], butterfly optimiza-



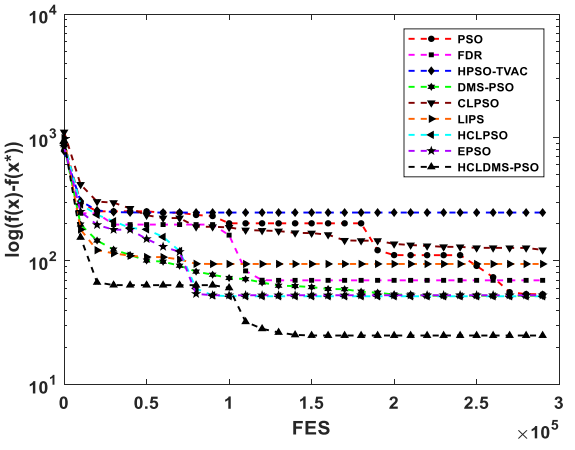
(a) F1 Function



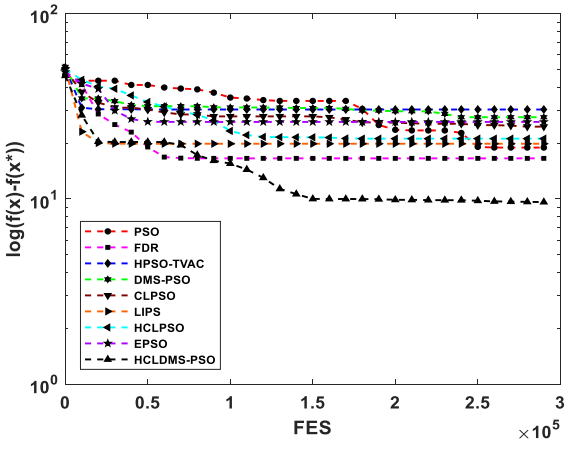
(b) F3 Function



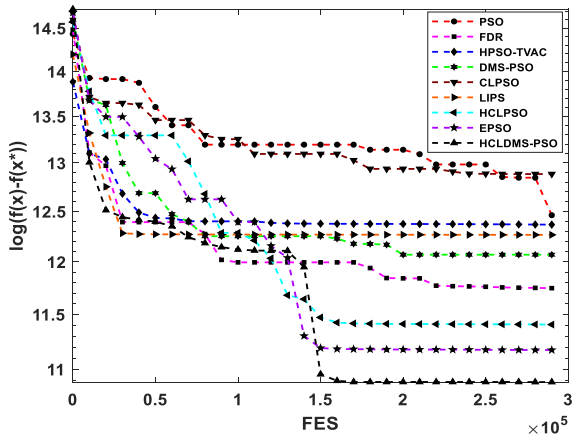
(c) F5 Function



(d) F10 Function

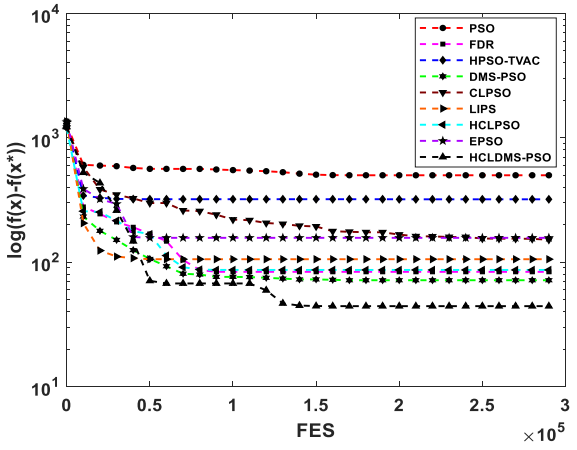


(e) F11 Function

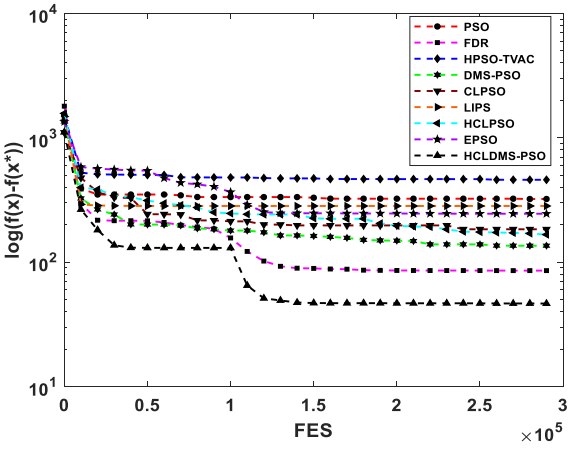


(f) F14 Function

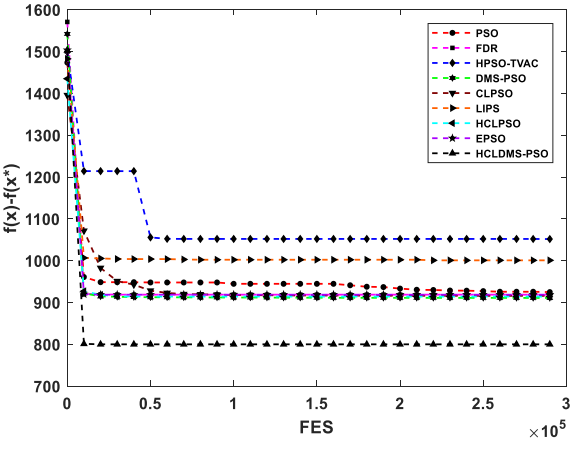
Fig. 3. Median convergence characteristics graphs of 30 dimensional CEC 2005 benchmark functions: (a) F1: shifted sphere function, (b) F3: Shifted Rotated High Conditioned Elliptic Function (c) F5: schwefel's problem2.6 with global optimum on bounds, (d) F10: shifted rotated Rastrigin's function, (e) F11: shifted rotated Weierstrass function, (f) F14: Shifted Rotated Expanded Scaffer's F6.



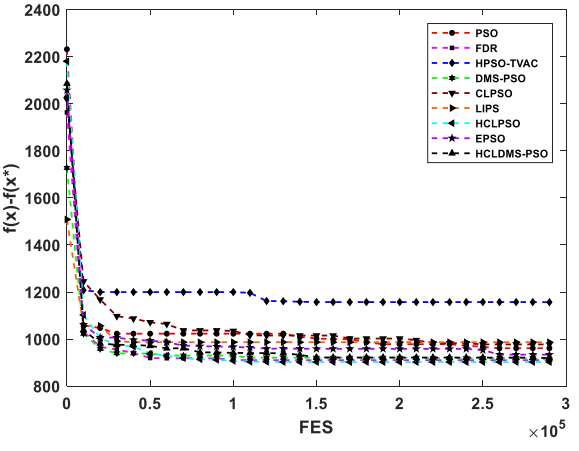
(a) F16 Function



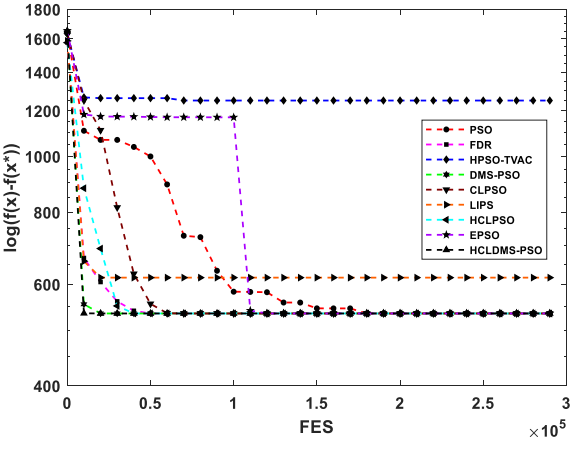
(b) F17 Function



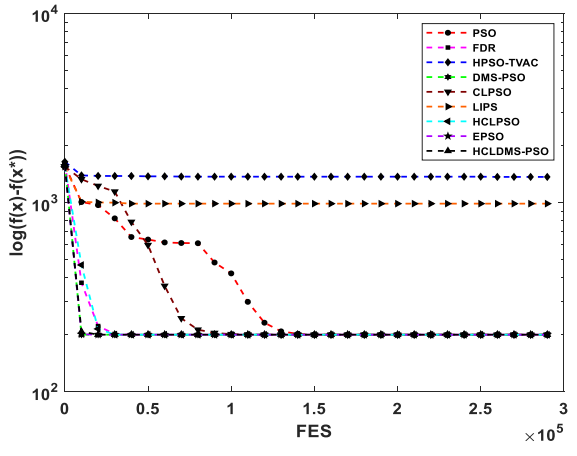
(c) F19 Function



(d) F22 Function



(e) F23 Function



(f) F25 Function

Fig. 4. Median convergence characteristics graphs of 30 dimensional CEC 2005 benchmark functions: (a) F16: rotated hybrid composition function, (b) F17: rotated hybrid composition function with noise in fitness, (c) F19: Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum, (d) F22: rotated hybrid composition function with high condition number matrix, (e) F23: Non-Continuous Rotated Hybrid Composition Function, (f) F25: rotated hybrid composition function without bounds.

Table 9
Key to comparative methods for CEC2017 benchmark functions.

Key	Method name	Year	References
BA	A New Metaheuristic Bat-Inspired Algorithm	2010	[49]
GWO	Grey Wolf Optimizer	2014	[25]
BOA	Butterfly Optimization Algorithm	2018	[1]
WOA	The Whale Optimization Algorithm	2016	[26]
MFO	Moth-flame Optimization Algorithm	2015	[24]
L-SHADE	Improving the Search Performance of SHADE Using Linear Population Size Reduction	2014	[38]
ABC	Artificial Bee Colony algorithm	2007	[14]
(G/T/L/E) ABC	Natural selection methods for artificial bee colony with new versions of onlooker bee	2018	[3]

tion algorithm (BOA) [1], whale optimization algorithm (WOA) [26], moth flame optimization (MFO) [24], L-SHADE (CEC2014 champion algorithm) [38], and artificial bee colony (ABC) [3,14]) on the CEC2017 Special Session on Real-Parameter Single Objective Optimization benchmark suite [2]. The CEC2017 test suite (Appendix 1) consists of 30 test functions. For all problems, the search range is $[-100, 100]^D$. In addition, F1-F3 are unimodal functions, and F2 has been excluded because it shows an unstable behavior, particularly for higher dimensions [2]. Moreover, F4-F10 are simple multimodal functions, F11-F20 are hybrid functions, and finally, F21-F30 are composition functions, which combine multiple test problems into a complex landscape.

The population size of the other state-of-the-art PSO variants (CLPSO, DMS-PSO, HCLPSO, and EPSO) and the proposed HCLDMS-PSO algorithm is set to 40, and is set to 100 for the other 11 evolutionary algorithms. For the other parameters not listed here, please refer to the original study. We use the 29 CEC 2017 benchmark test suite functions with the same parameter settings, i.e., 31 run times, an FES of 300,000, and 30 dimensions. The mean errors in the results obtained in the CEC2017 test suite are shown in Table 10, and the ranks of the mean performance are listed in Table 11. The Wilcoxon signed rank test results of the CEC2017 test suite with a significance level of $\alpha = 0.05$ for the 30-dimensional problems are listed in Table 12.

For the CEC2017 test suite, L-SHADE and EPSO achieve the best performance on unimodal functions F1 and F3, respectively. For the basic multimodal functions, the proposed HCLDMS-PSO achieves the best performance on F6, F7, and F9 and second best on F5 and F8. L-SHADE achieves the best performance on F5, F8, and F10, and TABC acquires the best performance on F4. For the F6 benchmark function, ABC/GABC/LABC/EABC/CLPSO/DMS-PSO/HCLPSO also obtain the same best mean error results, as indicated in Table 10. For the hybrid functions, the proposed HCLDMS-PSO obtains the best performance on F11 and F16, whereas L-SHADE achieves the best performance on F12-F14 and F18. In addition, CLPSO generates the best results on F15 and F19, and DMS-PSO achieves the best performance on F20. For the composition functions F21-F30, the proposed HCLDMS-PSO achieves the best performance on F22 and F29. TABC obtains the best performance on F24 and F26, and L-SHADE demonstrates the best performance on F23 and F30. Moreover, CLPSO, ABC, DMS-PSO, and EPSO generate the best results on F21, F25, F27, and F28, respectively.

According to the average and final ranks, L-SHADE (winner of CEC2014) achieves the best performance and the top rank. The proposed HCLDMS-PSO algorithm ranks second. HCLPSO and EPSO achieve third and fourth place, respectively. Although L-SHADE is slightly better than the proposed HCLDMS-PSO in some of the test functions, there is little difference between them. However, the results of L-SHADE on unimodal functions F1 are significantly different from those of the other evolutional algorithms. The mean error of L-SHADE is zero, whereas the errors of the other evolutional algorithms are no smaller than 1.08E+01. The reason behind this phenomenon is complicated and worth its own detailed investigation.

The Wilcoxon signed-rank test results in Table 12 show a comparison with four state-of-the-art PSO variants, in which the proposed HCLDMS-PSO wins on 17, 17, 11, and 14 benchmark functions, respectively. It is notable that the proposed HCLDMS-PSO is superior to some well-known meta heuristic algorithms, including BA, GWO, BOA, WOA, MFO, ABC, GABC, TABC, LABC, and EABC. The proposed HCLDMS-PSO only wins on ten benchmark functions compared with L-SHADE. Because the natural difference in the PSO variants and L-SHADE is a reduction of the population size, a future investigation into the PSO variants with a reduction in the population size is worth pursuing.

4.4. HCLDMS-PSO performance for real-world problem

The performance of HCLDMS-PSO was also evaluated using a widely applied real-world optimization problem known as wireless sensor networks (WSNs) coverage control [44]. WSNs can be adapted to numerous applications, such as target tracking, disaster warning, and wearable devices. In addition, they are large-scale and high-density networks that typically have an overlap in their coverage area. Moreover, a random deployment of sensor nodes cannot fully guarantee coverage of the sensing area, which leads to coverage holes in the WSNs. Thus, coverage control plays an important role in WSNs, influencing the network topology and energy consumption.

Suppose N sensor nodes deployed randomly in a two-dimensional target monitoring area, which can be viewed as a node set $G = \{n_1, n_2, n_3, \cdots, n_N\}$, where n_i denotes node i. The coordinate of node i is (x_i, y_i) and its sensing radius is r_i. If point

Table 10
The mean errors results obtained in CEC2017 test suite for problem size of 30 variables.

	F1	F3	F4	F5	F6	F7	F8	F9	F10	F11
BA	7.29E+10	2.16E+05	2.09E+04	5.08E+02	1.05E+02	1.46E+03	4.29E+02	2.13E+04	8.79E+03	2.44E+04
GWO	1.05E+09	2.94E+04	1.51E+02	8.66E+01	4.04E+00	1.61E+02	7.65E+01	5.39E+02	2.77E+03	4.75E+02
BOA	3.00E+10	6.67E+04	2.52E+03	3.25E+02	6.36E+01	5.14E+02	2.93E+02	6.89E+03	7.71E+03	2.96E+03
WOA	2.14E+06	1.57E+05	1.49E+02	2.66E+02	6.62E+01	5.06E+02	1.93E+02	7.67E+03	4.82E+03	3.55E+02
MFO	8.08E+09	7.74E+04	5.13E+02	1.75E+02	2.52E+01	3.51E+02	1.72E+02	5.05E+03	4.11E+03	2.37E+03
L-SHADE	**0.00E+00**	9.39E+03	4.14E+01	**2.46E+01**	1.43E−01	6.57E+01	**2.76E+01**	2.71E+01	**1.93E+03**	9.41E+01
ABC	1.32E+02	1.24E+05	3.36E+01	8.82E+01	**0.00E+00**	1.02E+02	8.85E+01	8.20E+02	2.31E+03	4.27E+02
GABC	2.86E+02	1.29E+05	2.75E+01	5.36E+01	**0.00E+00**	7.90E+01	6.35E+01	1.65E+02	2.05E+03	4.72E+02
TABC	3.35E+02	9.85E+04	**2.25E+01**	7.01E+01	1.62E−07	8.96E+01	7.30E+01	2.87E+02	2.03E+03	2.23E+02
LABC	4.48E+02	1.35E+05	3.01E+01	5.65E+01	**0.00E+00**	7.82E+01	6.16E+01	1.81E+02	2.09E+03	7.89E+02
EABC	4.73E+02	1.20E+05	3.41E+01	5.47E+01	**0.00E+00**	7.80E+01	6.10E+01	1.63E+02	2.15E+03	4.31E+02
CLPSO	1.08E+01	1.90E+04	5.12E+01	4.42E+01	**0.00E+00**	8.30E+01	5.10E+01	5.54E+01	2.14E+03	6.13E+01
DMS-PSO	2.34E+03	2.23E+03	6.19E+01	4.64E+01	**0.00E+00**	9.25E+01	4.83E+01	1.20E−01	2.92E+03	3.72E+01
HCLPSO	1.25E+02	1.08E−03	6.84E+01	4.13E+01	**0.00E+00**	8.58E+01	4.32E+01	1.06E+01	2.11E+03	6.90E+01
EPSO	3.01E+02	**8.82E−05**	2.52E+01	4.97E+01	4.47E−09	8.60E+01	5.01E+01	4.56E+00	2.11E+03	4.52E+01
HCLDMS-PSO	6.97E+02	2.84E+01	5.91E+01	2.75E+01	**0.00E+00**	**5.76E+01**	2.85E+01	**7.88E−02**	2.38E+03	**3.36E+01**
	F12	**F13**	**F14**	**F15**	**F16**	**F17**	**F18**	**F19**	**F20**	**F21**
BA	1.46E+10	1.29E+10	1.20E+07	1.83E+09	4.77E+03	4.63E+03	1.52E+08	2.16E+09	1.39E+03	7.15E+02
GWO	5.15E+07	9.17E+05	2.37E+05	2.09E+05	6.97E+02	2.40E+02	5.96E+05	1.31E+06	3.35E+02	2.75E+02
BOA	3.83E+09	1.53E+09	6.45E+05	4.48E+07	2.24E+03	8.54E+02	5.53E+06	6.38E+07	8.55E+02	4.09E+02
WOA	4.47E+07	1.47E+05	6.42E+05	6.59E+04	1.70E+03	8.83E+02	2.61E+06	2.39E+06	7.55E+02	4.51E+02
MFO	1.13E+08	4.45E+07	1.59E+05	3.55E+04	1.52E+03	6.64E+02	2.30E+06	1.84E+06	5.53E+02	3.81E+02
L-SHADE	**4.61E+03**	**2.35E+02**	**1.17E+02**	2.11E+02	4.47E+02	1.12E+02	**2.79E+02**	1.24E+02	1.62E+02	2.28E+02
ABC	4.09E+05	6.69E+03	6.34E+04	1.03E+03	6.82E+02	2.25E+02	1.60E+05	1.19E+03	2.64E+02	2.54E+02
GABC	5.28E+05	1.13E+04	9.91E+04	2.59E+03	5.63E+02	1.78E+02	1.97E+05	3.58E+03	1.89E+02	2.44E+02
TABC	4.78E+05	7.09E+03	9.63E+04	2.06E+03	6.03E+02	1.93E+02	1.79E+05	2.50E+03	2.44E+02	2.51E+02
LABC	6.40E+05	1.46E+04	1.15E+05	2.61E+03	6.12E+02	1.75E+02	2.24E+05	2.80E+03	2.33E+02	2.51E+02
EABC	4.25E+05	1.23E+04	6.01E+04	3.23E+03	5.93E+02	2.08E+02	1.53E+05	2.82E+03	2.14E+02	2.45E+02
CLPSO	3.28E+05	2.86E+02	2.29E+04	**1.21E+02**	4.98E+02	1.32E+02	1.21E+05	**5.28E+01**	1.58E+02	**2.21E+02**
DMS-PSO	1.65E+05	1.06E+04	5.62E+03	5.03E+03	4.46E+02	**6.76E+01**	1.19E+05	3.58E+03	**1.38E+02**	2.48E+02
HCLPSO	2.90E+04	7.23E+02	5.78E+03	2.98E+02	5.50E+02	1.21E+02	6.99E+04	1.36E+02	1.55E+02	2.36E+02
EPSO	2.92E+04	6.04E+02	3.83E+03	2.37E+02	5.14E+02	1.17E+02	1.02E+05	1.49E+02	1.91E+02	2.49E+02
HCLDMS-PSO	6.25E+04	6.79E+03	2.58E+03	8.42E+02	**3.41E+02**	7.96E+01	8.28E+04	1.71E+03	1.61E+02	2.28E+02
	F22	**F23**	**F24**	**F25**	**F26**	**F27**	**F28**	**F29**	**F30**	
BA	8.82E+03	1.45E+03	1.70E+03	5.30E+03	1.03E+04	2.17E+03	6.51E+03	6.63E+03	1.60E+09	
GWO	1.87E+03	4.35E+02	4.94E+02	4.57E+02	1.80E+03	5.28E+02	5.58E+02	7.55E+02	5.43E+06	
BOA	1.33E+03	6.74E+02	7.56E+02	1.14E+03	4.09E+03	6.48E+02	1.67E+03	1.99E+03	1.48E+08	
WOA	3.86E+03	7.41E+02	7.36E+02	4.53E+02	4.56E+03	6.74E+02	5.05E+02	1.90E+03	9.93E+06	
MFO	3.49E+03	5.19E+02	5.66E+02	7.60E+02	2.99E+03	5.38E+02	9.38E+02	1.06E+03	2.98E+05	
L-SHADE	1.72E+02	**3.77E+02**	4.48E+02	3.88E+02	1.36E+03	5.13E+02	3.53E+02	5.38E+02	**2.32E+03**	
ABC	4.94E+02	4.12E+02	4.65E+02	**3.84E+02**	3.48E+02	5.14E+02	4.00E+02	6.30E+02	5.58E+03	
GABC	4.90E+02	4.09E+02	4.86E+02	3.85E+02	9.22E+02	5.12E+02	4.06E+02	5.70E+02	7.69E+03	
TABC	5.12E+02	4.08E+02	**4.09E+02**	3.85E+02	**2.53E+02**	5.12E+02	4.01E+02	5.83E+02	6.97E+03	
LABC	3.35E+02	4.09E+02	4.78E+02	3.85E+02	6.41E+02	5.12E+02	4.10E+02	5.69E+02	7.65E+03	
EABC	3.45E+02	4.08E+02	4.90E+02	3.85E+02	9.43E+02	5.11E+02	4.06E+02	5.65E+02	7.62E+03	
CLPSO	1.30E+02	4.00E+02	4.39E+02	3.86E+02	7.02E+02	5.10E+02	4.16E+02	5.31E+02	5.40E+03	
DMS-PSO	1.00E+02	3.91E+02	4.59E+02	3.88E+02	1.45E+03	**5.04E+02**	3.34E+02	5.69E+02	4.77E+03	
HCLPSO	1.01E+02	3.98E+02	4.43E+02	3.87E+02	3.52E+02	5.15E+02	3.49E+02	4.98E+02	3.64E+03	
EPSO	1.72E+02	3.93E+02	4.62E+02	3.86E+02	3.27E+02	5.12E+02	**3.07E+02**	5.38E+02	3.86E+03	
HCLDMS-PSO	**1.00E+02**	3.79E+02	4.49E+02	3.87E+02	1.25E+03	5.07E+02	3.79E+02	**4.88E+02**	3.76E+03	

$P(x_p, y_p)$ is within the sensing radius of node n_i, we assume that it is covered by node n_i. The sensing rate $p(n_i, P)$ for pixel P covered by n_i is as follows:

$$p(n_i, P) = \begin{cases} 1, & d(n_i, P) \leqslant r_i \\ 0, & d(n_i, P) > r_i \end{cases} \tag{14}$$

The distance between pixel P and node n_i is $d(n_i, P) = \sqrt{(x_i - x_p)^2 + (y_i - y_p)^2}$. Because pixel P can be covered by several nodes at the same time, we consider whether pixel P is covered by node set G:

$$p(G, P) = 1 - \prod_{n_i \in N} [1 - p(n_i, P)] (i = 1, 2, 3, \cdots, N) \tag{15}$$

Finally, the coverage rate of the target area (i.e., the fitness function of this optimization problem) is formulated as follows:

Table 11
Rank of mean performance on CEC2017 test suite.

Functions	BA	GWO	BOA	WOA	MFO	L-SHADE	ABC	GABC	TABC	LABC	EABC	CLPSO	DMS-PSO	HCLPSO	EPSO	HCLDMS-PSO
F1	16	13	15	12	14	**1**	4	5	7	8	9	2	11	3	6	10
F3	16	7	8	15	9	5	12	13	10	14	11	6	4	2	**1**	3
F4	16	13	15	12	14	7	5	3	**1**	4	6	8	10	11	2	9
F5	16	11	15	14	13	**1**	12	7	10	9	8	4	5	3	6	2
F6	9	5	7	8	6	4	**1**	**1**	3	**1**	**1**	**1**	**1**	**1**	2	**1**
F7	16	12	15	14	13	2	11	5	9	4	3	6	10	7	8	**1**
F8	16	11	15	14	13	**1**	12	9	10	8	7	6	4	3	5	2
F9	16	11	14	15	13	5	12	8	10	9	7	6	2	4	3	**1**
F10	16	11	15	14	13	**1**	9	3	2	4	8	7	12	5	6	10
F11	16	12	15	8	14	6	9	11	7	13	10	4	2	5	3	**1**
F12	16	13	15	12	14	**1**	7	10	9	11	8	6	5	2	3	4
F13	16	13	15	12	14	**1**	5	9	7	11	10	2	8	4	3	6
F14	16	13	15	14	12	**1**	8	10	9	11	7	6	4	5	3	2
F15	16	14	15	13	12	2	6	8	7	9	10	**1**	11	4	3	5
F16	16	12	15	14	13	3	11	7	9	10	8	4	2	6	5	**1**
F17	16	12	14	15	13	3	11	8	9	7	10	6	1	5	4	2
F18	16	12	15	14	13	**1**	8	10	9	11	7	6	5	2	4	3
F19	16	12	15	14	13	2	5	11	7	8	9	**1**	10	3	4	6
F20	16	12	15	14	13	5	11	6	10	9	8	3	**1**	2	7	4
F21	16	12	14	15	13	2	11	5	10	9	6	**1**	7	4	8	3
F22	16	13	12	15	14	6	10	9	11	7	8	4	2	3	5	**1**
F23	16	12	14	15	13	**1**	11	10	8	9	7	6	3	5	4	2
F24	16	12	15	14	13	4	8	10	**1**	9	11	2	6	3	7	5
F25	16	13	15	12	14	10	**1**	4	3	2	5	7	11	9	6	8
F26	16	12	14	15	13	10	3	7	**1**	5	8	6	11	4	2	9
F27	16	12	14	15	13	9	10	8	6	5	4	3	**1**	11	7	2
F28	16	13	15	12	14	4	6	9	7	10	8	11	2	3	**1**	5
F29	16	12	15	14	13	5	11	9	10	7	6	3	8	2	4	**1**
F30	16	13	15	14	12	**1**	7	11	8	10	9	6	5	2	4	3
Average rank	15.76	11.83	14.17	13.41	12.79	**3.59**	8.17	7.79	7.24	8.07	7.55	4.62	5.66	4.24	4.34	**3.86**
Final rank	16	12	15	14	13	**1**	11	9	7	10	8	5	6	3	4	**2**
Best/2nd Best/Worst	0/0/29	0/0/0	0/0/0	0/0/0	0/0/0	10/4/0	2/0/0	1/0/0	3/1/0	1/1/0	1/0/0	4/3/0	5/5/0	1/6/0	2/3/0	7/6/0

Table 12
Wilcoxon signed rank test results of CEC2017 test suite with a significance level of $\alpha = 0.05$ for 30 dimensional problems.

Functions	Pairwise comparison HCLDMS-PSO versus														
	BA	GWO	BOA	WOA	MFO	L-SHADE	ABC	GABC	TABC	LABC	EABC	CLPSO	DMS-PSO	HCLPSO	EPSO
F1	+	+	+	+	+	–	–	–	–	0	0	–	0	–	–
F3	+	+	+	+	+	+	+	+	+	+	+	+	+	–	–
F4	+	+	+	+	+	–	–	–	–	–	–	0	0	0	–
F5	+	+	+	+	+	–	+	+	+	+	+	+	+	+	+
F6	+	+	+	+	+	+	0	0	+	0	0	0	0	0	+
F7	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
F8	+	+	+	+	+	0	+	+	+	+	+	+	+	+	+
F9	+	+	+	+	+	+	+	+	+	+	+	+	0	+	+
F10	+	+	+	+	+	–	0	–	–	–	–	–	+	–	–
F11	+	+	+	+	+	+	+	+	+	+	+	+	0	+	0
F12	+	+	+	+	+	–	+	+	+	+	+	+	+	–	–
F13	+	+	+	+	+	–	0	+	0	+	0	–	0	–	–
F14	+	+	+	+	+	–	+	+	+	+	+	+	+	+	+
F15	+	+	+	+	+	+	+	+	+	+	+	–	+	–	–
F16	+	+	+	+	+	0	+	+	+	+	+	+	+	+	+
F17	+	+	+	+	+	0	+	+	+	+	+	+	0	+	+
F18	+	+	+	+	+	–	+	+	+	+	+	+	+	0	0
F19	+	+	+	+	+	–	0	+	0	0	0	–	+	–	–
F20	+	+	+	+	+	0	+	0	+	+	+	0	–	–	0
F21	+	+	+	+	+	0	0	+	+	+	+	0	+	+	+
F22	+	+	+	+	+	+	+	+	+	+	+	+	0	0	+
F23	+	+	+	+	+	0	+	+	+	+	+	+	+	+	+
F24	+	+	+	+	+	0	0	+	0	0	+	0	+	–	+
F25	+	+	+	+	+	0	–	–	–	–	–	–	0	0	–
F26	+	+	+	+	+	+	–	–	–	–	–	–	+	–	–
F27	+	+	+	+	+	+	+	+	+	+	+	+	–	+	+
F28	+	+	+	+	+	0	0	0	0	+	0	+	–	–	–
F29	+	+	+	+	+	+	+	+	+	+	+	+	+	0	+
F30	+	+	+	+	+	–	+	+	+	+	+	+	+	0	0
+/0/–	29/0/0	29/0/0	29/0/0	29/0/0	29/0/0	10/9/10	18/7/4	21/3/5	20/4/5	21/4/4	20/5/4	17/5/7	17/9/3	11/7/11	14/4/11

$$MaximumF(x) = \frac{\sum_{P \in j \times k} p(G,P)}{j \times k} \tag{16}$$

where j and k denote the length and width of the target area. Swarm intelligence algorithms have been shown to be an effective approach to optimize node deployment and enhance network performance.

In this section, the sensor node N is set to 15, namely, a 30-dimensional WSNs coverage control problem. The size of the target area is $100 \times 100\,m^2$, and the sensing radius r_i is 15 m. All 16 EA algorithms in Section 4.3 were applied to this experiment. The run times and maximum FEs were set to 30 and 300,000, respectively.

As shown in Table 13, although the proposed HCLDMS-PSO is inferior to the winning algorithm of CEC2014, i.e., L-SHADE, and close in performance to TABC and EPSO, it performs better than the three other state-of-the-art PSO variants and nine evolutionary algorithms. As shown in Fig. 5, the proposed HCLDMS-PSO algorithm maintains the highest convergence speed during the early stage and slows down during the late stage. All experiment results suggest that the proposed HCLDMS-PSO algorithm is not only valid on the benchmark functions, but also applicable to solving real-world engineering problems.

Table 13
Comparison on the 30D WSNs coverage control problem.

algorithm	BA	GWO	BOA	WOA	MFO	L-SHADE	ABC	GABC
mean	0.7406	0.915	0.753	0.870	0.913	**0.931**	0.916	0.915
std	1.77E−02	9.73E−03	1.90E−02	2.15E−02	1.45E−02	**1.63E−03**	2.71E−03	5.65E−03
rank	16	9	15	14	10	**1**	6	8
algorithm	TABC	LABC	EABC	CLPSO	DMS-PSO	HCLPSO	EPSO	HCLDMS-PSO
mean	0.927	0.910	0.915	0.897	0.903	0.922	0.926	0.925
std	3.97E−03	2.30E−03	2.39E−03	6.26E−03	6.14E−03	1.08E−02	3.55E−03	5.32E−03
rank	2	11	7	13	12	5	3	4

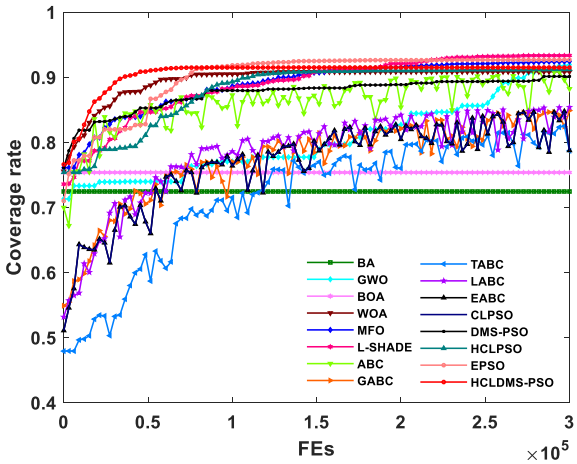


Fig. 5. Convergence curve of 30D WSNs coverage control problem.

4.5. Discussions

The experiment results of the two classical benchmark test suite functions and a real-world engineering problem verified the performance of the proposed HCLDMS-PSO algorithm in terms of convergence accuracy, convergence speed, and reliability. According the statistical results, the proposed HCLDMS-PSO algorithm executes better than the other existing state-of-the-art PSO approaches and various other meta-heuristic evolutionary algorithms on more unimodal, multimodal, expanded, and hybrid composition problems. However, for the CEC2005 test suite, the proposed HCLDMS-PSO algorithm does not perform well on two multimodal functions, F6 and F9, or on two expanded and hybrid composition functions, F13 and F15. For the CEC2017 test suite, the proposed HCLDMS-PSO algorithm does not perform well on unimodal function F1, two simple multimodal functions F4 and F10, and two composition functions F25 and F26. This can be explained through the "no free lunch" theorem of optimization [46], in which "any elevated performance over one class of problems is offset by the performance over another class." Therefore, a general-purpose universal optimization algorithm is theoretically impossible. No strategies can be expected to outperform any other on every type of optimization problem. The proposed HCLDMS-PSO algorithm can obtain either the best or second-best results on most of the test 10-dimensional or 30-dimensional problems, however. It was verified that the proposed HCLDMS-PSO algorithm benefits from the modified CLPSO and DMS-PSO algorithms described in Section 3.1 and enhances the universality and robustness of the performance.

Furthermore, a more intuitive measurement of the exploitation and exploration capabilities of the two subpopulations of the proposed HCLDMS-PSO algorithm, we compared the diversities of the CL subpopulation, the DMS subpopulation, and the whole population of HCLDMS-PSO. It can be clearly seen from the graphs on the diversity characteristic that the diversity of the CL subpopulation decreases quickly and is mainly responsible for the exploitation. The DMS subpopulation maintains the population diversity throughout the evolutionary search process and enables it to avoid a premature convergence. The incorporation between the CL and DMS subpopulations enables HCLDMS-PSO to achieve a trade-off between the exploration and exploitation capability and perform better than other state-of-the-art PSO variants for both unimodal and multimodal functions. In addition, the computational complexity and cost of the proposed HCLDMS-PSO algorithm are comparable to those of the other state-of-the-art PSOs.

The proposed HCLDMS-PSO algorithm uses two heterogeneous subpopulations that are allocated to enhance the exploitation and exploration capabilities without crippling one another. First, we select the CL strategy with the *gbest* of the whole population as the exploitation subpopulation of the proposed HCLDMS-PSO algorithm, which has been verified to achieved a good performance in terms of the HCLPSO algorithm's exploitation subpopulation [22] and can avoid both the "oscillation" and the "two steps forward, one step back" phenomena. Second, although a DMS strategy was selected as an exploration subpopulation, the original DMS strategy still incurs certain drawbacks, as described in Section 2.3. We overcome them by classifying the search level of all sub-swarms and constructing a novel nonlinear adaptive decreasing inertia weight formula, and by introducing a non-uniform mutation operator. The modified DMS strategy can effectively avoid a premature convergence and improve the exploration capability of the DMS subpopulation particles. Finally, the *gbest* of the whole population applies a Gaussian mutation operator to help it jump out of the local optimum.

5. Conclusions

In this study, the proposed HCLDMS-PSO divided the whole population into two heterogeneous subpopulations, similarly to HCLPSO for exploitation and exploration. The exploitation subpopulation exemplar employed a CL strategy guided by the

gbest of the whole population, which is consistent with the exploitation subpopulation in the HCLPSO algorithm. However, the exploration subpopulation in HCLDMS-PSO is specially designed through a modified DMS strategy rather than a CL strategy in HCLPSO. In the canonical DMS strategy, it is a disadvantage for different sub-swarms to use the same linear decreasing inertia weight parameter. We proposed classifying the search levels of the DMS subpopulation sub-swarms and constructed a novel nonlinear adaptive decreasing inertia weight for different sub-swarms, introducing a non-uniform mutation operator to enhance the exploration capability. Finally, the *gbest* of the whole population also adopted a Gaussian mutation operator to avoid falling into the local optimum. The particles of the two subpopulations will update their velocity independently without crippling each other to prevent a loss of diversity.

The effect of the different tuning parameters on the performance of the proposed HCLDMS-PSO algorithms was first studied using different values of the 25 CEC2005 test suite functions. A performance comparison was then conducted using the CEC2005 and CEC2017 benchmark optimization functions and a real-world WSNs coverage control problem. The results showed that the proposed HCLDMS-PSO algorithms offer improvements terms of the convergence speed, accuracy, and reliability on most optimization problems as compared with other existing state-of-the-art PSOs and various other metaheuristic evolutionary algorithms. It is worth noting that the performance of the proposed HCLDMS-PSO for a few unimodal functions and a few multimodal functions is unsatisfying. The proposed HCLDMS-PSO algorithm achieved slightly worse results than L-SHADE (winning algorithm of CEC2014) for the CEC2017 test suite and a real-world application problem. The reason behind this is complicated and there must be some underlying mechanisms worthy of a future detailed investigation. To address this issue, integrating a population size reduction and other excellent learning or dynamic cluster strategies may play an important role in improving the PSO and will therefore be investigated in the future. Our follow-up studies will also include the application of the proposed optimization algorithm to other complex practical engineering problems.

CRediT authorship contribution statement

Shengliang Wang: Conceptualization, Methodology, Software, Writing - original draft. **Genyou Liu:** Supervision, Project administration. **Ming Gao:** Software, Resources. **Shilong Cao:** Software, Resources. **Aizhi Guo:** Writing - review & editing. **Jiachen Wang:** Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was jointly supported by the National Key Research Program of China "Collaborative Precision Positioning Project" Grant (No.2016YFB0501900), the National Natural Science Foundation of China (Grant No. 41774017; 41621091). We are grateful to Prof. P. N. Suganthan from the School of Electrical & Electronics Engineering, Nanyang Technological University, Singapore for code on their homepages (http://www3.ntu.edu.sg/home/epnsugan/).

Appendix 1. CEC2017 test suite

Function type	Test Functions	Search Range	Category	$F(x^*)$
Unimodal Functions	F1: Shifted and Rotated Bent Cigar Function	$[-100, 100]^D$	UN	100
	F2: Shifted and Rotated Sum of Different Power Function*	$[-100, 100]^D$	UN	200
	F3: Shifted and Rotated Zakharov Function	$[-100, 100]^D$	UN	300
Simple Multimodal Functions	F4: Shifted and Rotated Rosenbrock's Function	$[-100, 100]^D$	MN	400
	F5: Shifted and Rotated Rastrigin's Function	$[-100, 100]^D$	MN	500
	F6: Shifted and Rotated Expanded Scaffer's F6 Function	$[-100, 100]^D$	MN	600
	F7: Shifted and Rotated Lunacek Bi_Rastrigin Function	$[-100, 100]^D$	MN	700
	F8: Shifted and Rotated Non-Continuous Rastrigin's Function	$[-100, 100]^D$	MN	800
	F9: Shifted and Rotated Levy Function	$[-100, 100]^D$	MN	900

(*continued on next page*)

Appendix 1 (*continued*)

Function type	Test Functions	Search Range	Category	$F(x^*)$
	F10: Shifted and Rotated Schwefel's Function	$[-100, 100]^D$	MN	1000
Hybrid Functions	F11: Hybrid Function 1 (N = 3)	$[-100, 100]^D$	H	1100
	F12: Hybrid Function 2 (N = 3)	$[-100, 100]^D$	H	1200
	F13: Hybrid Function 3 (N = 3)	$[-100, 100]^D$	H	1300
	F14: Hybrid Function 4 (N = 4)	$[-100, 100]^D$	H	1400
	F15: Hybrid Function 5 (N = 4)	$[-100, 100]^D$	H	1500
	F16: Hybrid Function 6 (N = 4)	$[-100, 100]^D$	H	1600
	F17: Hybrid Function 6 (N = 5)	$[-100, 100]^D$	H	1700
	F18: Hybrid Function 6 (N = 5)	$[-100, 100]^D$	H	1800
	F19: Hybrid Function 6 (N = 5)	$[-100, 100]^D$	H	1900
	F20: Hybrid Function 6 (N = 6)	$[-100, 100]^D$	H	2000
Composition Functions	F21: Composition Function 1 (N = 3)	$[-100, 100]^D$	C	2100
	F22: Composition Function 2 (N = 3)	$[-100, 100]^D$	C	2200
	F23: Composition Function 3 (N = 4)	$[-100, 100]^D$	C	2300
	F24: Composition Function 4 (N = 4)	$[-100, 100]^D$	C	2400
	F25: Composition Function 5 (N = 5)	$[-100, 100]^D$	C	2500
	F26: Composition Function 6 (N = 5)	$[-100, 100]^D$	C	2600
	F27: Composition Function 7 (N = 6)	$[-100, 100]^D$	C	2700
	F28: Composition Function 8 (N = 6)	$[-100, 100]^D$	C	2800
	F29: Composition Function 9 (N = 3)	$[-100, 100]^D$	C	2900
	F30: Composition Function 10 (N = 3)	$[-100, 100]^D$	C	3000

U Unimodal, *M* multimodal, *N* non-separable, *H* hybrid (unimodal or multimodal), *C* composition;
*F2 has been excluded because it shows unstable behavior especially for higher dimensions.

References

[1] S. Arora, S. Singh, Butterfly optimization algorithm: a novel approach for global optimization, Soft Comput. 23 (3) (2018) 715–734.
[2] N.H. Awad, M.Z. Ali, P.N. Suganthan, J.J. Liang, B.Y. Qu, Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, 2016, Tech. Rep.
[3] M.A. Awadallah, M.A. Al-Betar, A.L.A. Bolaji, E.M. Alsukhni, H. Al-Zoubi, Natural selection methods for artificial bee colony with new versions of onlooker bee, Soft Comput. 23 (15) (2018) 6455–6494.
[4] W.-N. Chen, J. Zhang, Y. Lin, N. Chen, Z.-H. Zhan, H.S.-H. Chung, Y. Li, Y.-H. Shi, Particle swarm optimization with an aging leader and challengers, IEEE Trans. Evol. Comput. 17 (2) (2012) 241–258.
[5] Y. Chen, L. Li, H. Peng, J. Xiao, Y. Yang, Y. Shi, Particle swarm optimizer with two differential mutation, Appl. Soft Comput. 61 (2017) 314–330.
[6] J. Cheng, Y. Li, Optimization of structural support locations using a hybrid genetic algorithm, in: Proceedings of the 2013 IEEE Symposium on Computational Intelligence for Engineering Solutions, IEEE, 2013, pp. 1–6.
[7] R. Cheng, Y. Jin, A social learning particle swarm optimization algorithm for scalable optimization, Inform. Sci. 291 (2015) 43–60.
[8] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evol. Comput. 6 (1) (2002) 58–73.
[9] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.
[10] Y.-J. Gong, J. Zhang, H.S.-H. Chung, W.-N. Chen, Z.-H. Zhan, Y. Li, Y.-H. Shi, An efficient resource allocation scheme using particle swarm optimization, IEEE Trans. Evol. Comput. 16 (6) (2012) 801–816.
[11] S.-Y. Ho, H.-S. Lin, W.-H. Liauh, S.-J. Ho, OPSO: Orthogonal particle swarm optimization and its application to task assignment problems, IEEE Trans. Syst. Man Cybern. Part A Syst. Humans 38 (2) (2008) 288–298.
[12] A. Iosifidis, A. Tefas, I. Pitas, On the kernel extreme learning machine classifier, Pattern Recognit. Lett. 54 (2015) 11–17.
[13] J. Jie, W. Wang, C. Liu, B. Hou, Multi-swarm particle swarm optimization based on mixed search behavior, in: Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications, 2010, pp. 605–610.
[14] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (3) (2007) 459–471.
[15] C. Li, S. Yang, T.T. Nguyen, A self-learning particle swarm optimizer for global optimization problems, IEEE Trans. Syst. Man Cybern. Part B Cybern. 42 (3) (2011) 627–646.
[16] Z. Li, W. Wang, Y. Yan, Z. Li, PS–ABC: a hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems, Expert Syst. Appl. 42 (22) (2015) 8881–8895.
[17] J.-J. Liang, P.N. Suganthan, Dynamic multi-swarm particle swarm optimizer, in: Proceedings of the IEEE Swarm Intelligence Symposium, IEEE, 2005, pp. 124–129.
[18] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295.
[19] W.H. Lim, N.A.M. Isa, An adaptive two-layer particle swarm optimization with elitist learning strategy, Inform. Sci. 273 (2014) 49–72.

[20] W.H. Lim, N.A.M. Isa, Particle swarm optimization with increasing topology connectivity, Eng. Appl. Artif. Intell. 27 (2014) 80–102.
[21] N. Lynn, P.N. Suganthan, Ensemble particle swarm optimizer, Appl. Soft Comput. 55 (2017) 533–548.
[22] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm Evol. Comput. 24 (2015) 11–24.
[23] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler, maybe better, IEEE Trans. Evol. Comput. 8 (3) (2004) 204–210.
[24] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249.
[25] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey Wolf Optimizer, Adv. Eng. Software 69 (2014) 46–61.
[26] S. Mirjalili, A. Lewis, The Whale Optimization Algorithm, Adv. Eng. Software 95 (2016) 51–67.
[27] M. Nasir, S. Das, D. Maity, S. Sengupta, U. Halder, P.N. Suganthan, A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization, Inform. Sci. 209 (2012) 16–36.
[28] M.S. Nobile, P. Cazzaniga, D. Besozzi, R. Colombo, G. Mauri, G. Pasi, Fuzzy Self-Tuning PSO: A settings-free algorithm for global optimization, Swarm Evol. Comput. 39 (2018) 70–85.
[29] A. Passaro, A. Starita, Particle swarm optimization for multimodal functions: a clustering approach, J. Artif. Evol. Appl. 2008 (2008) 8.
[30] T. Peram, K. Veeramachaneni, C.K. Mohan, Fitness-distance-ratio based particle swarm optimization, in: Proceedings of the 2003 IEEE Swarm Intelligence Symposium, IEEE, 2003, pp. 174–181.
[31] B.-Y. Qu, P.N. Suganthan, S. Das, A distance-based locally informed particle swarm model for multimodal optimization, IEEE Trans. Evol. Comput. 17 (3) (2012) 387–402.
[32] J.K.R. 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.
[33] 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.
[34] A. Sedki, D. Ouazar, Hybrid particle swarm optimization and differential evolution for optimal design of water distribution systems, Adv. Eng. Inf. 26 (3) (2012) 582–591.
[35] M. Setayesh, M. Zhang, M. Johnston, A novel particle swarm optimisation approach to detecting continuous, thin and smooth edges in noisy images, Inform. Sci. 246 (2013) 28–51.
[36] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: Proceedings of the IEEE Congress on Evolutionary Computation, IEEE, 1998, pp. 69–73.
[37] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.-P. Chen, A. Auger, S. Tiwari, Problem definitions and evaluation criteria for the CEC special session on real-parameter optimization, Proc. IEEE Congress Evol. Comput. 2005 (2005) 1–50.
[38] R. Tanabe, A.S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2014, pp. 1658–1665.
[39] M.R. Tanweer, S. Suresh, N. Sundararajan, Dynamic mentoring and self-regulation based particle swarm optimization algorithm for solving complex real-world optimization problems, Inform. Sci. 326 (2016) 1–24.
[40] M.R. Tanweer, S. Suresh, N. Sundararajan, Self regulating particle swarm optimization algorithm, Inform. Sci. 294 (2015) 182–202.
[41] D. Tian, Z. Shi, MPSO: Modified particle swarm optimization and its applications, Swarm Evol. Comput. 41 (2018) 49–68.
[42] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, M.J.I.S. Ventresca, Enhancing particle swarm optimization using generalized opposition-based learning, Inform. Sci. 181 (20) (2011) 4699–4714.
[43] H. Wang, Y. Jin, J. Doherty, Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems, IEEE Trans. Cybern. 47 (9) (2017) 2664–2677.
[44] J. Wang, C. Ju, Y. Gao, A.K. Sangaiah, G.-J. Kim, A PSO based energy efficient coverage control algorithm for wireless sensor networks, Comput. Mater. Contin. 56 (3) (2018) 433–446.
[45] L. Wen, Z. Xi, The research of PSO algorithms with non-linear time-decreasing inertia weight, in: Proceedings of the 7th World Congress on Intelligent Control and Automation, IEEE, 2008, pp. 4002–4005.
[46] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[47] G. Xu, Q. Cui, X. Shi, H. Ge, Z.-H. Zhan, H.P. Lee, Y. Liang, R. Tai, C. Wu, Particle swarm optimization based on dimensional learning strategy, Swarm Evol. Comput. 45 (2019) 33–51.
[48] G. Xu, An adaptive parameter tuning of particle swarm optimization algorithm, Appl. Math. Comput. 219 (9) (2013) 4560–4569.
[49] X.-S. Yang, A new metaheuristic bat-inspired algorithm, Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer, 2010.
[50] Z.-H. Zhan, J. Zhang, Y. Li, H.S.-H. Chung, Adaptive particle swarm optimization, IEEE Trans. Syst. Man Cybern. Part B Cybern. 39 (6) (2009) 1362–1381.