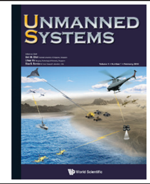
Unmanned Systems, Vol. 14, No. 1 (2026) 95–117

DOI: 10.1142/S2301385025500840

Unmanned Systems
https://www.worldscientific.com/worldscinet/us



A Particle Swarm Optimizer with Biased Exploration and Exploitation for High-Dimensional Optimization and Feature Selection

Minchong Chen [*,§], Hong Li [*], Jiwei Tu [*], Bin He [*], Zhen Yin [*], Xuejing Hou [†], Qi Yu [†,‡]

[*] *College of Electronic and Information Engineering, Tongji University, 4800 Cao'an Road, Shanghai, P. R. China*

[†] *College of Transportation Engineering, Tongji University, 4800 Cao'an Road, Shanghai, P. R. China*

Autonomous intelligent systems have been widely implemented in a broad range of applications. These applications often involve data-dense tasks where an efficient feature selection process is required to eliminate redundant features and improve model performance. However, the feature selection tasks with high dimensionality still remain challenging to deal with. In order to address high-dimensional feature selection problems with greater effectiveness and efficiency, this paper proposes a particle swarm optimizer variant named PSO-BEE that allows the swarm to take full advantage of exploration and exploitation at due evolutionary stages. At the early stage, a large size of candidate exemplar group is formed for diversity enhancement, attempting to find as many new combinations of features as possible. At the later stage, a tiny size of candidate exemplar group is constructed, allowing updated particles to learn from the few best elite exemplars to continuously refine feature selection solutions with a lower classification error rate. Three PSO-BEE variants with distinct preferences for exploration and exploitation are proposed based on different parameter adjustment strategies. Experiments are executed on 500, 1000, and 2000-dimensional benchmark suites presented by *CEC* and real-world feature selection datasets from the *Machine Learning Repository*. Experimental results demonstrate the competitive performance of PSO-BEE in high-dimensional global optimization and feature selection when compared with several state-of-the-art approaches.

Keywords: Feature selection; high dimensionality; particle swarm optimizer; candidate exemplar group; global optimization.

1. Introduction

Autonomous intelligent systems have been widely implemented in a broad range of applications such as medical diagnosis [1, 2], infrastructure inspection, and autonomous driving. These applications often involve data-dense tasks like image segmentation [3], visual tracking [4, 5], and path planning [6] where an efficient feature selection process is required to eliminate redundant features and improve model performance. However, the high-dimensional feature selection problems still remain challenging to be tackled, a class of problems that have been proven to be NP-hard [7]. The main difficulties of high-dimensional feature selection tasks are listed as follows:

- **The complexity of datasets.** The complexity of the dataset increases with the rise in its dimensionality and size. There may be more noise or interrelated features in datasets with higher dimensionality, and it is quite difficult to distinguish whether a feature is useful or redundant without prior knowledge of a dataset.
- **The curse of dimensionality** [8]. As the dimensionality of a dataset increases, the number of possible combinations of features increases exponentially. Searching for the most suitable feature selection solution as being considerably hard like "the search of a needle in a haystack" [9].

Received 8 August 2024; Revised 21 October 2024; Accepted 21 October 2024; Published 29 November 2024. This paper was recommended for publication in its revised form by editorial board member, Jinqiang Cui.
Email Addresses: [‡]*2233440@tongji.edu.cn*, [§]*Swain_1002@163.com*
[‡]Corresponding author.

For datasets with a large number of features, many of them are irrelevant and redundant. The existence of these redundant features not only increases the computational resources required for classification but also worsens the classification accuracy of classifiers that are trained on them. Therefore, it is expected to distinguish and remove as many redundant features as possible while retaining useful ones.

Many researchers have struggled to utilize heuristic algorithms to evaluate and remove redundant features of datasets. Due to the easy implementation of canonical PSO (cPSO) [10], it has been widely used to cope with such tasks [11]. However, cPSO shows difficulty in achieving satisfactory performance on feature selection problems with high dimensionality, mainly attributes to its premature convergence in high-dimensional search spaces. To improve the performance of cPSO on high-dimensional feature selection, many works have been done by proposing PSO variants based on novel swarm topology or learning strategies. For example, the pairwise grouping topology and competitive learning mechanism designed in CSO [12, 13]; the stochastic dominant learning strategy proposed in SDLSO [14]; a bi-directional feature fixation framework devised in BBPSO-BDFF [15]. These PSO variants significantly improve the swarm diversity and effectively avoid premature convergence, thus being able to find feature selection solutions with better quality.

Other heuristic algorithms besides PSO are also utilized as wrappers to tackle feature selection problems [16]. Including but not limited to ant colony optimization (ACO)-based approaches [17], genetic algorithm (GA)-based algorithms [18], and differential evolution (DE)-based approaches [19]. Compared to PSO variants, these algorithms are better at balancing convergence and diversity during the optimization of feature selection solutions. However, heuristic algorithms still suffer from "the curse of dimensionality", and there is still room for improvement in diversity preservation and convergence acceleration on high-dimensional optimization and feature selection problems.

In order to take full advantage of exploration and exploitation in high-dimensional environment, this paper proposes a particle swarm optimizer with biased exploration and exploitation (PSO-BEE), which is employed to address large-scale global optimization and serves as a wrapper to deal with high-dimensional feature selection problems. The main contributions of this paper are listed as follows:

- A general swarm dynamic segmentation strategy is proposed, which aims at taking full advantage of exploration and exploitation and is parameter-free. After utilizing a threshold $\boldsymbol{T}$ to segment the swarm into candidate exemplar group (***CEG***, the group where exemplars are selected) and candidate updater group (***CUG***, the group where updated particles are selected), updated particles in ***CUG*** explore and exploit the search space by following the guidance of two exemplars that are randomly selected from ***CEG***.
- Three PSO-BEE variants are proposed based on three different changing trends of $\boldsymbol{T}$. The three PSO-BEE variants have different preferences for exploitation and exploration and can therefore be utilized to solve high-dimensional optimization problems with different characteristics. Besides, the effects of $\boldsymbol{T}$ on swarm diversity are verified mathematically and experimentally.
- Three PSO-BEE variants are tested on representative benchmark functions to verify the effectiveness of the designed strategies in regulating the swarm diversity and convergence characteristics. The one with highest Friedman ranking are employed to address five real-world high-dimensional feature selection problems. The proposed algorithm demonstrates competitive performance when compared with several state-of-the-art approaches.

The rest contents of this paper are organized as follows. Section 2 briefly introduces the PSO and some heuristic algorithms that serve as wrappers to address feature selection tasks. Thereafter, in Sec. 3, the devise of PSO-BEE is elaborated. The effects of the threshold segmentation strategy adopted are analyzed from a mathematical perspective. Following that is the experimental study in Sec. 4, which contains the experiments of PSO-BEE on large-scale optimization benchmark suites and real-world feature selection datasets. Conclusions and future work are given in Sec. 5.

2. Literature Review

Without loss of generality, the optimization problems are defined as follows:

$$\min f(\boldsymbol{x}) \quad \boldsymbol{x} = [x^1, x^2, \ldots, x^D] \in \boldsymbol{R}^D, \tag{1}$$

where D ($D \geq 500$) [20] is the number of decision variables of bounded vector $\boldsymbol{x}$, $f(\boldsymbol{x})$ is the objective function to be optimized. In this paper, $f(\boldsymbol{x})$ of feature selection tasks is defined as the classification error rate of classifiers trained with selected features.

2.1. *Particle swarm optimization*

Kennedy *et al.* [10] drew inspiration from the foraging behavior of bird flocks and first propose particle swarm optimization (PSO). In the canonical PSO (cPSO), the velocity and position of particles are updated according to

(2) and (3):

$$v_i(t+1) = \omega v_i(t) + c_1 r_1(\text{pbest}_i(t) - x_i(t)) + c_2 r_2(\text{gbest}(t) - x_i(t)), \quad (2)$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad (3)$$

where $v_i(t)$ and $x_i(t)$ are the velocity and position of the ith particle at the tth generation. gbest is the best position discovered by the current whole swarm and pbest_i is the historically best position of the ith particle. As for parameters, ω is the inertia weight, c_1 and c_2 are acceleration coefficients utilized to adjust the influence of pbest_i and gbest, respectively. r_1 and r_2 are two random numbers within (0,1) adopted to enrich the search behavior.

cPSO shows good performance in low-dimensional optimization problems due to its fast convergence speed. However, cPSO is found to easily encounter premature convergence or be trapped in locally optimal areas when it is applied in a high-dimensional environment. Both phenomena should be attributed to the global dominance of gbest, an exemplar whose evolutionary information is shared by all updated particles, which leads to poor swarm diversity and seriously damages the exploration ability of the swarm.

2.2. *Heuristic algorithms for feature selection*

In the feature selection problem, the wrapper method is a strategy to evaluate and select features by model performance. It treats feature selection as a search problem, trains the model with different feature subsets, and selects the best feature combination based on the performance of the model (such as classification accuracy). In recent years, much work has been done around designing heuristic algorithm wrappers to address feature selection and classification problems.

Gu *et al.* [13] propose a competitive swarm optimizer (CSO) for high-dimensional feature selection tasks. In this algorithm, particles are randomly grouped in pairs. The fitness loser in each group searches for better feature combinations by learning from the corresponding winner and the mean position of the swarm, while the winner directly enters the next generation for diversity preservation. Yang *et al.* [14] propose an adaptive stochastic dominant learning swarm optimizer (SDLSO) and employ it on feature selection problems. Updated particles conduct exploration and exploitation on features by following the guidance of two different exemplars that are stochastically selected. Yang *et al.* [15] propose a bi-directional feature fixation (BDFF) framework for PSO to address feature selection problems, where updated particles move toward two opposing search directions to adequately search for feature subsets with different sizes. Zhu *et al.* [21] propose a binary restructuring PSO named BRPSO. BRPSO merely depends on the differences between values derived from the position updating formula and a random number to update particles, allowing the algorithm to more effectively adapt to the discrete optimization problem of feature selection. Song *et al.* [22] propose a bare bones PSO (BBPSO) with mutual information. In this algorithm, two local search operators are designed to improve exploitation performance, and an adaptive flip mutation operator is devised for jumping out of locally optimal feature selection solutions. Chen *et al.* [23] propose BP-PSO, which combines PSO with a back propagation (BP) neural network. During the initialization, a chaotic model is added to improve swarm diversity. Then an adaptive factor is designed to encourage the swarm to search for new feature combinations. Recently, Tijjani *et al.* [24] propose an enhanced binary particle swarm optimization (EBPSO) for feature selection tasks. This approach determines the probability of the particles updating using position instead of velocity so as to effectively avoid premature convergence.

For heuristic algorithms other than PSO variants to serve as wrappers to solve feature selection tasks, Zhou *et al.* [17] propose a random following ant colony optimization (RFACO). The random following strategy designed is utilized to strengthen the communication between colony members, encouraging the swarm to escape from locally optimal areas. Deng *et al.* [18] propose a feature thresholds guided genetic algorithm (FTGGA). The feature threshold, which continuously evolves to direct the genetic algorithm to refine better solutions, represents the probability of a corresponding feature being selected. Wang *et al.* [19] propose a diversity-based multi-objective differential evolution approach (DMBDE). In order to enhance swarm diversity, the feature selection solution with a higher diversity score will be preferred. Awadallah *et al.* propose four novel swarm intelligence algorithm for feature selection tasks. They introduce Lévy flight and opposition-based learning strategies into artificial rabbit optimization (ARO) and propose several ARO variants based on different transfer functions [25]. In order to balance exploration and exploitation in white shark optimization (WSO), they propose WSO variants with binary adaptation, comprehensive learning and heterogeneous WSO [26], respectively. Motivated by the survival behavior of horse swarms and the hunting behavior of rat swarms, they further propose several enhanced binary horse herd optimization (HOA) [27] and rat swarm optimization (RSO) algorithms [28], which achieve promising performance on feature selection problems with various dimensions.

2.3. *Discussion on reviewed literature*

In this work, we pay attention to designing novel particle swarm optimizer variant to address high-dimensional

Table 1. Summary of the reviewed PSO variants for high-dimensional feature selection.

Alg. \ Char.	Hybrid algorithm	Self-adaptive strategy	Convergence acceleration	Diversity enhancement
CSO [13]				✓
SDLSO [14]		✓	✓	
BDFF-PSO [15]		✓	✓	✓
BRPSO [21]				✓
BBPSO [22]	✓		✓	
BP-PSO [23]	✓	✓		✓
EBPSO [24]				✓

global optimization and feature selection problems. Therefore, in this section, we briefly analyze the characteristics of reviewed PSO variants for feature selection tasks, while the analysis of heuristic algorithms other than PSO variants is omitted. The reviewed algorithms and their corresponding characteristics and listed in Table 1.

The algorithms listed in Table 1 have achieved satisfactory performance on high-dimensional feature selection problems; however, they are not able to take full advantage of the exploration and exploitation behaviors of the particle swarm. In CSO [13], the mean position of the swarm is shared by all updated particles, which could confine the diversity and exploration ability of the swarm. Besides, elite particles with better quality also learn from the mean level of the swarm, which damages the exploitation abilities of elite individuals. As for algorithms other than BDFF-PSO [15], they either focus on accelerating convergence to improve exploitation capabilities while ignoring diversity maintenance, or they concentrate on enhancing diversity to strengthen exploration while neglecting local area exploitation. Consequently, it is expected to take both exploration and exploitation into account and make full use of both to improve the optimization performance of an algorithm.

3. Proposed Method

3.1. *Discretization of continuous variables for feature selection*

In feature selection problems, each feature has only two states: selected or not selected. The decision variables in this problem are discrete. However, the proposed and other comparative algorithms are originally designed to solve continuous global optimization problems. Therefore, we utilize Algorithm 1 to convert the continuous decision variables to discrete ones. According to our investigation, when the range of decision variables is too small, all of the algorithms used converge too quickly. When the range is too large, it leads to redundancy. Consequently, we choose to use the same variable range settings as in the large-scale optimization experiments in this work [29, 30], thus setting the upper bound *ub* to 32. Namely, $x \in [-32, 32]$ is adopted in the experiments of all algorithms on feature selection problems in this paper. Since the variable value range is symmetric about 0 and each feature has an equal probability of being selected and being not selected, one feature is selected as its corresponding dimension is greater than 0.

3.2. *Methodology and framework of the proposed algorithm*

In swarm intelligence algorithms, swarm diversity and convergence are strongly coupled and interact with each other; diversity preservation and convergence acceleration are two conflicting goals. Since exploration relies on diversity preservation, while exploitation requires convergence pressure, exploration and exploitation are mutually restrained. In this case, neither exploration nor exploitation can be fully utilized. Therefore, it is expected to weaken the mutual constraints between the two search behaviors. To achieve this goal, we encourage the swarm to perform exploration without considering exploitation at the early stage and to conduct exploitation while neglecting exploration at the later stage.

Algorithm 1. Converting continuous values to discrete values for feature selection.

Input: Particle $x = [x^1, x^2, \ldots, x^D] \in [-ub, ub]^D$, D is the dimensionality, x^j is the j-th dimension of x.
Output: The selected feature subset S.

$S \leftarrow \emptyset$;
  for $j = 1$ to D **do**
    if $x^j > 0$ **do**
      $S \leftarrow S \cup \{j\}$;
    else
      continue;
    end if
  end for
return S;

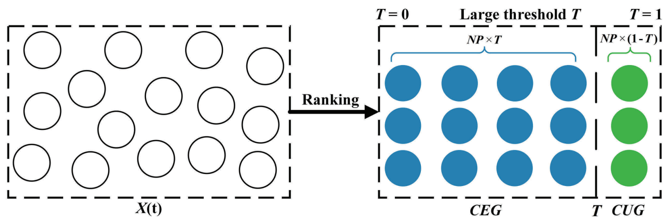


Fig. 1. Swarm exploration behavior.

The exploration and exploitation behaviors largely rely on the diversity and convergence characteristics of the swarm, which are deeply influenced by the exemplars that guide the evolution of updated particles. Before the swarm is convergent, the diversity of candidate exemplars is positively related to their number. The larger the candidate exemplar group, the better the diversity of the selected exemplars. On the contrary, the smaller the candidate exemplar group, the poorer the diversity of the selected exemplars. Consequently, the exploration and exploitation behaviors of the swarm can be controlled by the size of the candidate exemplar group. The exploration and exploitation behaviors of the swarm in our proposed algorithm are exhibited in Figs. 1 and 2, respectively. In Fig. 1, a large value of threshold $\boldsymbol{T}$ segments the swarm into a large size of ***CEG*** and a small size of ***CUG***, which diversifies the selection of exemplars and preserve swarm diversity to encourages the swarm to perform extensive exploration. In Fig. 2, a small value of threshold $\boldsymbol{T}$ segments the swarm into a small size of ***CEG*** and a large size of ***CUG***, which allows the updated particles to learn from the few best elite exemplars to accelerate convergence and conduct efficient exploitation.

Motivated by the above ideas, a dynamic particle swarm optimizer with biased exploration and exploitation (PSO-BEE) is proposed by dynamically changing the size of the candidate exemplar group (***CEG***) and the candidate updater group (***CUG***).

3.3. *PSO-BEE*

Given a swarm with NP particles, the key steps of the proposed PSO-BEE are elaborated as follows:

(1) Utilize a threshold $\boldsymbol{T}$ to segment the swarm into candidate exemplar group (***CEG***) and candidate updater group (***CUG***). The former is the group where exemplars are selected, and the latter is the group where updated particles are selected. The size of ***CEG*** and ***CUG*** is determined in (4) and (5):

$$S_{\boldsymbol{CEG}} = \mathrm{ceil}(\mathrm{NP} \times \boldsymbol{T}), \tag{4}$$

$$S_{\boldsymbol{CUG}} = \mathrm{NP} - S_{\boldsymbol{CEG}}, \tag{5}$$

where $S_{\boldsymbol{CEG}}$ and $S_{\boldsymbol{CUG}}$ are the sizes of ***CEG*** and ***CUG***, NP is the swarm size. All particles in ***CEG*** will be retained and directly enter the next generation. For each particle in ***CUG***, the probability of updating or being retained is both 0.5.

(2) When a particle x_i in ***CUG*** is updated, it learns from two different exemplars that are randomly selected from ***CEG***. In specific, the velocity and position of x_i are updated according to (6) and (7):

$$\begin{aligned} v_i(t+1) = r_1 v_i(t) + r_2(x_{e1}(t) - x_i(t)) \\ + \phi r_3(x_{e2}(t) - x_i(t)), \end{aligned} \tag{6}$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \tag{7}$$

where $v_i(t)$ and $x_i(t)$ are the velocity and position of the ith particle at the tth generation. x_{e1} and x_{e2} are two different exemplars that are randomly selected from ***CEG***. Note that x_{e1} is the one who has better fitness. $r_1 v_i$ is the inertia component for search stability, where r_1 is randomly valued within $(0, 1)$. r_2 and r_3 are also randomized within $(0, 1)$ to enrich the search behavior. ϕ is the parameter used to adjust the influence of x_{e2}.

(3) After one iteration of the swarm, the value of threshold $\boldsymbol{T}$ is updated according to (8):

$$\boldsymbol{T} = \boldsymbol{T}_{\max} - (\boldsymbol{T}_{\max} - \boldsymbol{T}_{\min}) \times \left(\frac{\mathrm{fes}}{\mathrm{Maxfes}}\right)^{\alpha}, \tag{8}$$

where $\boldsymbol{T}_{\max}$ and $\boldsymbol{T}_{\min}$ are the maximum and minimum values of $\boldsymbol{T}$, respectively. α is adopted to control the trend of $\boldsymbol{T}$. fes is the fitness evaluations already consumed before the current generation, Maxfes is the maximum fitness evaluation preset. Note that the value of $\boldsymbol{T}$ remains unchanged within one generation. For a clear comprehension, the pseudocode of PSO-BEE is outlined in Algorithm 2.

Note that in global optimization, the fitness represents the distance between the current solution and the global optimal solution in the objective space. In feature selection, the fitness is defined as the classification error rate of a classifier trained on given datasets.

In step (1), when the swarm size NP and the value of $\boldsymbol{T}$ are given, at the current generation, the number of individuals among the swarm that are being retained is

$$\mathrm{NP} \times \boldsymbol{T} + \frac{1}{2} \times \mathrm{NP} \times (1 - \boldsymbol{T}) = \frac{\mathrm{NP}(1 + \boldsymbol{T})}{2}, \tag{9}$$

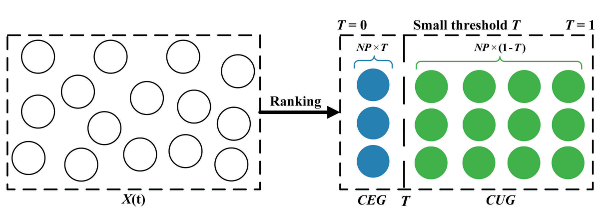


Fig. 2. Swarm exploitation behavior.

Algorithm 2. Pseudocode of PSO-BEE.

Input: Swarm size NP, maximum fitness evaluation Maxfes, control parameter ϕ.
Output: The final best solution x and its fitness $f(x)$.

Randomly initialize the swarm;
Calculate the fitness of each particle;
fes = NP;
while fes < Maxfes **do**
  Sort particles according to fitness;
  Calculate $\boldsymbol{T}$ according to (8);
  Segment the swarm into $\boldsymbol{CEG}$ and $\boldsymbol{CUG}$ by $\boldsymbol{T}$;
  for $i = \text{ceil}(\text{NP} \times \boldsymbol{T}) + 1$ to NP **do**
    if $rand \geq 0.5$ **do**
      Stochastically select two different exemplars in $\boldsymbol{CEG}$: x_{e1}, x_{e2};
      if $f(x_{e2}) \leq f(x_{e1})$ **do**
        Swap $e1$ and $e2$;
      end if
      Update the velocity and position of x_i according to (6) and (7);
      fes = fes + 1;
    end if
  end for
  Calculate the fitness of each updated particle;
end while
Return output;

correspondingly, the number of updated particles among the swarm is

$$\text{NP} - \frac{\text{NP}(1+\boldsymbol{T})}{2} = \frac{\text{NP}(1-\boldsymbol{T})}{2}. \tag{10}$$

The value of threshold $\boldsymbol{T}$ is larger at the early stage, it can be learned from (9) that, at this period, most particles among the swarm are retained without any update, which is beneficial to diversity preservation. As the evolution procedure advances, the value of threshold $\boldsymbol{T}$ gradually becomes small. It can be inferred from (10) that more and more individuals among the swarm are participating in the update, which is to the advantage of convergence acceleration and exploitation.

In step (2), the fitness relationship of x_{e1}, x_{e2} and x_i satisfies $f(x_{e1}) \leq f(x_{e2}) \leq f(x_i)$. Each updated individual only learns from particles that have better fitness. Better-fit exemplars possess promising evolutionary information, which qualifies them to guide the learning of inferior ones. This learning mechanism will reduce the computational resources wasted in poor areas [31]. Besides, the randomness during the selection of exemplars plays an important role in diversity preservation.

In step (3), first, $\boldsymbol{T}_{\max}$ and $\boldsymbol{T}_{\min}$ in (8) cannot be, respectively, set to 1 and 0, because once $\boldsymbol{T}$ equals 1, the size of ***CUG*** is 0, which means that no particle will participate in the update. Correspondingly, when the value of $\boldsymbol{T}$ is 0, the size of ***CEG*** is 0, which means that no particle can be selected as an exemplar. In order to avoid the occurrence of the above phenomena and excessive parameter adjustments, $\boldsymbol{T}_{\max}$ is set to a number that is large enough and very close to 1, while $\boldsymbol{T}_{\min}$ is set to a number that is small enough and very close to 0, namely, $\boldsymbol{T}_{\max} = 0.99$, $\boldsymbol{T}_{\min} = 0.01$. Enter the corresponding values into each variable, (8) can be arranged as (11)

$$\boldsymbol{T} = 0.99 - 0.98 \times \left(\frac{\text{fes}}{\text{Maxfes}}\right)^{\alpha}. \tag{11}$$

Second, given the swarm size NP and threshold $\boldsymbol{T}$, the probability of one certain candidate exemplar in ***CEG*** being selected by one certain updated particle in ***CUG*** is calculated according to (12). The operation of simultaneously selecting two different exemplars for each updated particle can be regarded as two samplings without replacement in the classical model of probability:

$$p(\boldsymbol{T}) = 1 - \frac{C^2_{(\text{NP}\times\boldsymbol{T})-1}}{C^2_{\text{NP}\times\boldsymbol{T}}} = \frac{2}{\text{NP}\times\boldsymbol{T}}. \tag{12}$$

This probability can also be regarded as the number of times that one certain candidate exemplar is selected by one certain updated particle in its update. Then the mean number of times that one certain candidate exemplar is selected by all updated particles in their update within a generation is calculated according to (13):

$$\begin{aligned} n(\boldsymbol{T}) &= \frac{2}{\text{NP}\times\boldsymbol{T}} \times \frac{1}{2} \times \text{NP} \times (1-\boldsymbol{T}) \\ &= \frac{1}{\boldsymbol{T}} - 1. \end{aligned} \tag{13}$$

Based on (13), it can be seen that the larger the value of $\boldsymbol{T}$, the fewer times the same candidate exemplar in ***CEG*** is selected by other updated individuals; the more diverse the selection of exemplars, which is conducive to swarm diversity maintenance. On the contrary, the smaller the value of $\boldsymbol{T}$, the more times the same candidate exemplar in ***CEG*** is chosen by other updated particles; the more intensive the selection of exemplars, which is beneficial to swarm convergence acceleration. Based on the dynamic change strategy of $\boldsymbol{T}$ formulated in (11), the value range of $n(\boldsymbol{T})$ should be [0.01,99]. Consequently, the threshold $\boldsymbol{T}$ has a great impact on the swarm diversity characteristics. The detailed discussions of the effects of $\boldsymbol{T}$ are given in Sec. 4.2.3.

In summary, in PSO-BEE, the threshold $\boldsymbol{T}$ significantly affects the swarm diversity characteristics, it thus becomes the key parameter used to control the swarm exploration and exploitation behaviors.

3.4. *Differences between PSO-BEE and some PSO variants*

Compared with two representative PSO variants: TPLSO [32], APSO-DEE [33], the main differences between PSO-BEE and them are, respectively, summarized as follows:

- In TPLSO, the main purpose of the mass learning phase (Phase 1) is exploration. However, the inferior and worst particles in each group have a shared exemplar; two-thirds of the particles among the swarm are participating in the update. Both of the above situations accelerate convergence to some extent, risking the swarm in declined diversity and not conducive to exploration. The elite learning phase (Phase 2) is mainly responsible for exploitation. Whereas, the selection of exemplars for inferior elite individuals is still diverse. While in the proposed algorithm, first, thanks to the large size of ***CEG*** and tiny size of ***CUG*** that formed at the early stage, the diversity preservation ability of PSO-BEE is extremely greater than that of TPLSO. Second, due to the tiny size of ***CEG*** constructed at the later stage, the selection of exemplars in PSO-BEE is far more centralized than that in TPLSO, endowing the former the higher exploitation efficiency.
- In APSO-DEE, a novel exploration and exploitation decoupled learning structure is devised. The essence of this algorithm is to quantify and control the diversity during the swarm convergence process. APSO-DEE achieves a direct balance between convergence and diversity by making individuals distributed as evenly as possible in the objective space. First, a local sparseness degree is utilized to measure the diversity at particle level. The introduction of a new metrics increases the complexity of the algorithm. Moreover, it takes additional computational resources and storage resources to calculate and store the local sparseness degree of each particle, and the additional resources consumed are positively related to the swarm size. While in PSO-BEE, no extra metrics are introduced; the only issue that requires additional computational resources is the update of threshold $\boldsymbol{T}$. Whereas, the additional resources required by $\boldsymbol{T}$ are independent of the swarm size and are negligible compared to those required by fitness. Second, the swarm at the end of evolution is preferred to be convergent in the most promising area. Instead of still being distributed evenly in the objective space or the decision space, particles should be gathered near the optimal areas found when the fitness evaluation is exhausted. Continuous mobilization of particles into areas with sparse swarm distribution could result in insufficient exploitation at the later stage. This situation is avoided to the greatest extent by the proposed algorithm.

Arguably, opposite to seeking balancing between exploration and exploitation or between diversity and convergence, an idea that LSOPs have been handled traditionally by most existing and ongoing swarm intelligence algorithms, this paper put forward with an innovative optimization mentality of taking full advantage of one side at the expense of the other at due evolutionary stages for improved search efficiency.

3.5. *Different versions of PSO-BEE*

In formula (8), by varying the value of α, the threshold $\boldsymbol{T}$ will have different changing trends. Specifically, when $\alpha > 1$, $\boldsymbol{T}$ shows a convex quadratic change; when $\alpha = 1$, $\boldsymbol{T}$ changes linearly; when $0 < \alpha < 1$, $\boldsymbol{T}$ shows a concave quadratic change. To make the work representative, we set α to 2, 1, and 0.5, respectively.

Substitute the values of α, three change strategies of $\boldsymbol{T}$ are formulated in (14)–(16), respectively. As depicted in Fig. 3, the threshold $\boldsymbol{T}$ is dynamically changed based on three different representative elementary functions, which are convex quadratic function (cv), linear function (l), and concave quadratic function (cc), respectively. Three versions of $\boldsymbol{T}$ decrease as the evolutionary process progresses, which meets the motivation of PSO-BEE:

$$\boldsymbol{T} = \boldsymbol{T}_{\max} - (\boldsymbol{T}_{\max} - \boldsymbol{T}_{\min}) \times \left(\frac{\text{fes}}{\text{Maxfes}}\right)^2, \tag{14}$$

$$\boldsymbol{T} = \boldsymbol{T}_{\max} - (\boldsymbol{T}_{\max} - \boldsymbol{T}_{\min}) \times \frac{\text{fes}}{\text{Maxfes}}, \tag{15}$$

$$\boldsymbol{T} = \boldsymbol{T}_{\max} - (\boldsymbol{T}_{\max} - \boldsymbol{T}_{\min}) \times \left(\frac{\text{fes}}{\text{Maxfes}}\right)^{0.5}. \tag{16}$$

Based on three different change strategies of $\boldsymbol{T}$, three different versions of PSO-BEE are proposed correspondingly, which are PSO-BEE with $\boldsymbol{T}$ varying as a convex

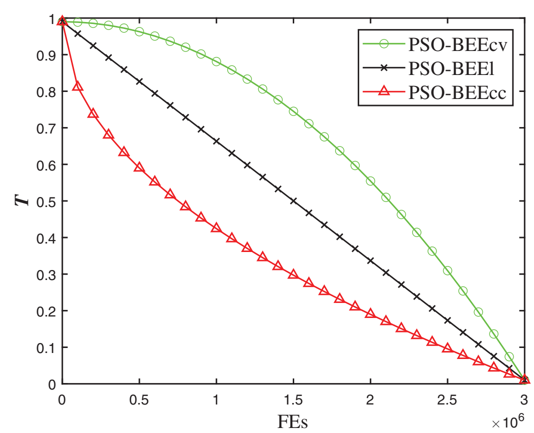


Fig. 3. The dynamic changing trends of $\boldsymbol{T}$ in three different versions of PSO-BEE.

quadratic function (PSO-BEEcv); PSO-BEE with ***T*** varying as a linear function (PSO-BEEl); and PSO-BEE with ***T*** varying as a concave quadratic function (PSO-BEEcc). The dynamic change trends of the thresholds in the three algorithms are depicted in the green line, black line and red line in Fig. 3, respectively.

For PSO-BEEcv, since ***T*** attenuates slowly at the early stage, it was not until at the later stage that it begins to attenuate rapidly, which keeps ***T*** at relatively large values for the majority of generations, allowing PSO-BEEcv to prolong the period of exploration while shortening the period of exploitation. Therefore, PSO-BEEcv is more suitable for problems that place higher demands on swarm diversity and exploration.

For PSO-BEEcc, on the contrary, the threshold ***T*** attenuates rapidly at the early stage and attenuates slowly at the later stage, which keeps ***T*** at relatively small values for the majority of generations, allowing PSO-BEEcc to shorten the period of exploration while prolonging the period of exploitation. Consequently, PSO-BEEcc is more fit for problems that raise higher requirements on swarm convergence and exploitation. For PSO-BEEl, generally, it is the average level between PSO-BEEcv and PSO-BEEcc.

From a mathematical perspective, it can be seen from Fig. 3 that, at any intermediate stage other than the beginning and end, the value of ***T*** in PSO-BEEcv is greater than that in PSO-BEEl, and the value of ***T*** in PSO-BEEl is greater than that in PSO-BEEcc. According to the analysis given in Sec. 3.3, before the swarm is convergent, the larger the value of threshold ***T***, the better the diversity of the swarm, the more conducive it is to exploration, and vice versa. Therefore, the swarm diversity and exploration ability of PSO-BEEcv, PSO-BEEl, and PSO-BEEcc should satisfy the relationship of $\triangle$PSO-BEEcv $>$ $\triangle$PSO-BEEl $>$ $\triangle$PSO-BEEcc. While the swarm convergence and exploitation ability should satisfy $\triangle$PSO-BEEcc $>$ $\triangle$PSO-BEEl $>$ $\triangle$PSO-BEEcv.

The comparison results and discussions of PSO-BEEcv, PSO-BEEl and PSO-BEEcc on solution quality and swarm diversity will be given in Sec. 4.2.1.

3.6. *Convergence condition analysis*

Convergence plays a significant role in the performance of swarm intelligence algorithms. In general, the convergence is analyzed without considering the search stagnation. This section investigates the convergence of $E(x(t))$ in PSO-BEE based on the theory given by [34], where $E(x(t))$ is the expectation of an arbitrary individual at the tth generation. The convergence proof of PSO-BEE is shown as follows.

The components of each dimension of position are updated independently, for simplicity, the update of an arbitrary particle at the tth generation can be rewritten as (17) and (18)

$$\begin{aligned} v(t+1) = {} & r_1(x(t) - x(t-1)) \\ & + r_2(x_{e1}(t) - x(t)) \\ & + \phi r_3(x_{e2}(t) - x(t)), \end{aligned} \tag{17}$$

$$x(t+1) = x(t) + v(t+1). \tag{18}$$

Based on (17) and (18), $x(t+1)$ can be replaced by (19), where $l = 1 + r_1 - r_2 - \phi r_3$,

$$\begin{aligned} x(t+1) = {} & lx(t) - r_1 x(t-1) \\ & + r_2 x_{e1}(t) + \phi r_3 x_{e2}(t). \end{aligned} \tag{19}$$

Then the mathematical expectation of $x(t+1)$ is exhibited in (20), where μ_{r_1}, μ_{r_2}, μ_{r_3}, $\mu_{x_{e1}}$, and $\mu_{x_{e2}}$ and are expectation values of according variables:

$$\begin{aligned} E(x(t+1)) = {} & E(l)E(x(t)) - \mu_{r_1} E(x(t-1)) \\ & + \mu_{r_2}\mu_{x_{e1}}(t) + \phi\mu_{r_3}\mu_{x_{e2}}(t). \end{aligned} \tag{20}$$

Based on (20), a recurrence relation between $E(x(t+1))$ and $E(x(t))$ can be arranged as (21), where $\mu = \mu_{r_2}\mu_{x_{e1}}(t) + \phi\mu_{r_3}\mu_{x_{e2}}(t)$,

$$\begin{bmatrix} E(x(t+1)) \\ E(x(t)) \end{bmatrix} = \begin{bmatrix} E(l) & -\mu_{r_1} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} E(x(t)) \\ E(x(t-1)) \end{bmatrix} + \begin{bmatrix} \mu \\ 0 \end{bmatrix}. \tag{21}$$

For simplicity, M is defined as (22)

$$M = \begin{bmatrix} E(l) & -\mu_{r_1} \\ 1 & 0 \end{bmatrix}. \tag{22}$$

According to the conclusion given by [34], the necessary and sufficient condition to ensure the convergence of $E(x(t))$ is that the magnitude of the eigenvalues of M, $|\lambda_1, \lambda_2| = |E(l) \pm \sqrt{E(l)^2 - 4\mu_{r_1}}/2|$, should be less than 1. Since r_1, r_2, and r_3 are all randomly generated within $(0, 1)$, μ_{r_1}, μ_{r_2}, and μ_{r_3} are $1/2$. Consequently, the necessary and sufficient condition for the convergence of $E(x(t))$ of an arbitrary particle can be arranged as (23)

$$\left| \frac{2 - \phi \pm \sqrt{\phi^2 - 4\phi - 4}}{4} \right| < 1. \tag{23}$$

By analyzing (23), the necessary and sufficient condition for the convergence of $E(x(t))$of an arbitrary particle is shown in (24):

$$-1 < \phi < 5. \tag{24}$$

Based on the analysis results above, $-1 < \phi < 5$ will be employed to ensure the convergence of $E(x(t))$ of an arbitrary particle in PSO-BEE.

3.7. *Complexity analysis*

With a given fitness evaluation, the time complexity of swarm intelligence algorithms is generally the extra time used in each generation without considering the time of the fitness calculation.

From Algorithm 2, we can see that PSO-BEE directly uses two exemplars in the current swarm to guide the exploration and exploitation of updated particles. Except for the function evaluation time, PSO-BEE takes $O(\mathrm{NP} \times D)$ computing time to update particles in each generation. Consequently, PSO-BEE has the same efficiency in time as cPSO [10], which also takes $O(\mathrm{NP} \times D)$ computing time in each generation.

As for the space complexity, in PSO-BEE, no storage is needed for storing the gbest and pbest. PSO-BEE only needs $O(\mathrm{NP} \times D)$ space to store the position of the swarm and another $O(\mathrm{NP} \times D)$ storage to keep the velocity. Consequently, compared with cPSO, PSO-BEE can save $O(\mathrm{NP} \times D)$ space.

In summary, PSO-BEE is as efficient as cPSO in terms of time but occupies fewer storage resources.

4. Experiments

4.1. *Experimental settings*

In the experimental section, in order to comprehensively analyze the performance of the proposed algorithm, we first compare the performance of the aforementioned three different versions of PSO-BEE from the perspectives of solution quality and swarm diversity. Next, the best of three different versions of PSO-BEE is chosen to compared with several state-of-the-art large-scale algorithms. Note that three different categories of optimizers are included in the compared algorithms: PSO variant, CC framework-based algorithm, and hybrid algorithm. Thereafter, ablation experiments are conducted to independently analyze the influence of different parameters on the proposed algorithm. After that, the scalability of the proposed algorithm is investigated by testing it on benchmark functions with different dimensionalities. Finally, the proposed algorithm is applied to tackle five high-dimensional feature selection datasets in the real world.

Experiments are executed on two most representative and widely used large-scale benchmark suites: CEC'2010 benchmark [29] and CEC'2013 benchmark [30]. The CEC'2010 benchmark contains 20 1000-*D* functions, which can be classified into three categories, fully separable, partially separable, and nonseparable. Each of these three categories of functions includes two features: unimodal and multimodal. As an extension of those functions in the CEC'2010 benchmark, the above three function categories are retained while a new category is added in the CEC'2013 benchmark: overlapping. There are only 15 functions contained in this benchmark, particularly, the dimensions of F_{13} and F_{14} are 905, while the dimensions of the rest of the functions are 1000. Basically, compared with CEC'2010 benchmark, CEC'2013 benchmark is more complex and more challenging to optimize.

Sections 4.2.1 and 4.2.3 in this paper involve quantitative analyses of swarm diversity characteristics. In our experiments, the swarm diversity is calculated by the standard deviation (Std), which is formulated in (25) and (26).

$$D(X) = \frac{1}{\mathrm{NP}} \sum_{i=1}^{\mathrm{NP}} \sqrt{\sum_{d=1}^{D} (x_i^d - \overline{x}^d)^2}, \tag{25}$$

$$\overline{x}^d = \frac{1}{\mathrm{NP}} \sum_{i=1}^{\mathrm{NP}} x_i^d, \tag{26}$$

where $D(X)$ is the measurement of the swarm diversity, $\overline{x}^d$ is the mean position of swarm X.

Unless otherwise stated, the maximum fitness evaluation Maxfes is set to $3000 \times D$ (D is the dimensionality). For fair comparisons, the mean value and standard deviation are utilized to measure the performance of different algorithms, which are obtained by applying each algorithm to each function for 25 independent runs. The best mean values obtained by different algorithms or different parameters with respect to each function are highlighted by a gray background. To tell the statistical significance, the Wilcoxon rank sum test is conducted at the significance level of $\alpha = 0.05$.

All the experiments we conducted are on MATLAB R2023a running on a PC with an Inter-Core i7-11400F CPU and the Microsoft Windows 11 Enterprise 64-bit operating system.

4.2. *Experimental results on large-scale benchmark suites*

4.2.1. *The investigation on three PSO-BEE variants*

In Sec. 3.5, three different versions of PSO-BEE are introduced based on three different dynamic change strategies of threshold $\boldsymbol{T}$. In this section, we investigate the performance of the three different PSO-BEE from the perspectives of solution quality and diversity characteristics. Experiments are executed on the CEC'2013 large-scale benchmark [30]. For fair comparisons, the same settings of NP and ϕ are used in the three versions, namely, $\mathrm{NP} = 500$, $\phi = 0.4$. Then, in Sec. 4.2.2, the best one is selected to compare with other state-of-the-art algorithms on both the CEC'2010 benchmark [29] and the CEC'2013 benchmark.

From Table 2, PSO-BEEcc achieves the best average results on 12 out of 15 functions and thus obtains the highest rank. The solutions attained by PSO-BEEcc on each function in the CEC'2010 benchmark and the CEC'2013 benchmark when the fitness evaluation counter reaches 1.2E+05, 6.0E+05, and 3.0E+06 are available in Tables S1

Table 2. Experimental results of three different versions of PSO-BEE on the CEC'2013 large-scale benchmark with 3E+06 fitness evaluations.

Function	PSO-BEEcv	PSO-BEEl	PSO-BEEcc
F_1	1.2918E−23	1.0895E−24	4.3916E−27
F_2	6.8837E+02	7.2007E+02	8.0073E+02
F_3	2.1598E+01	2.1595E+01	2.1598E+01
F_4	2.3242E+09	9.3365E+08	4.9128E+08
F_5	1.6682E+06	9.3664E+05	7.3913E+05
F_6	1.0607E+06	1.0609E+06	1.0603E+06
F_7	3.2398E+05	7.7788E+04	4.5433E+04
F_8	4.6986E+13	3.1291E+13	1.8167E+13
F_9	4.0587E+07	4.0011E+07	4.0699E+07
F_{10}	9.4017E+07	9.4042E+07	9.4008E+07
F_{11}	7.3845E+07	3.2161E+07	9.0602E+06
F_{12}	1.0373E+03	1.0404E+03	1.0286E+03
F_{13}	1.2355E+08	1.6515E+07	4.5403E+06
F_{14}	3.7073E+07	1.5841E+07	1.0947E+07
F_{15}	4.1900E+07	1.1619E+07	3.3620E+06
Friedman ranking	2.57	2.07	**1.37**

and S2 in the supplementary material, where the best, median, worst, mean value, and standard deviation of 25 independent runs are given. PSO-BEEl wins first place on F_3 and F_9 and gets the second rank. PSO-BEEcv achieves the best solution on F_2 and obtains the third rank. Besides, there are basically no significant differences between the solutions obtained by the three different versions of PSO-BEE on F_3, F_6, F_9, and F_{10}. As for the above experimental results, theoretical analysis is given from the perspectives of swarm diversity and convergence characteristics.

Figure 4 exhibits the diversity curves on the selected six functions (F_2, F_7, and F_{11} to F_{14}), where F_2 is fully-separable multimodal function, F_7 is partially-separable multimodal function, F_{11} is partially-separable unimodal function, F_{12} is overlapping multimodal functions, F_{13} and F_{14} are overlapping unimodal functions, respectively. The six functions singled out contain all features of the CEC'2013 benchmark. In order to make the contrast obvious, only the diversity curves of PSO-BEEcv and PSO-BEEcc are exhibited.

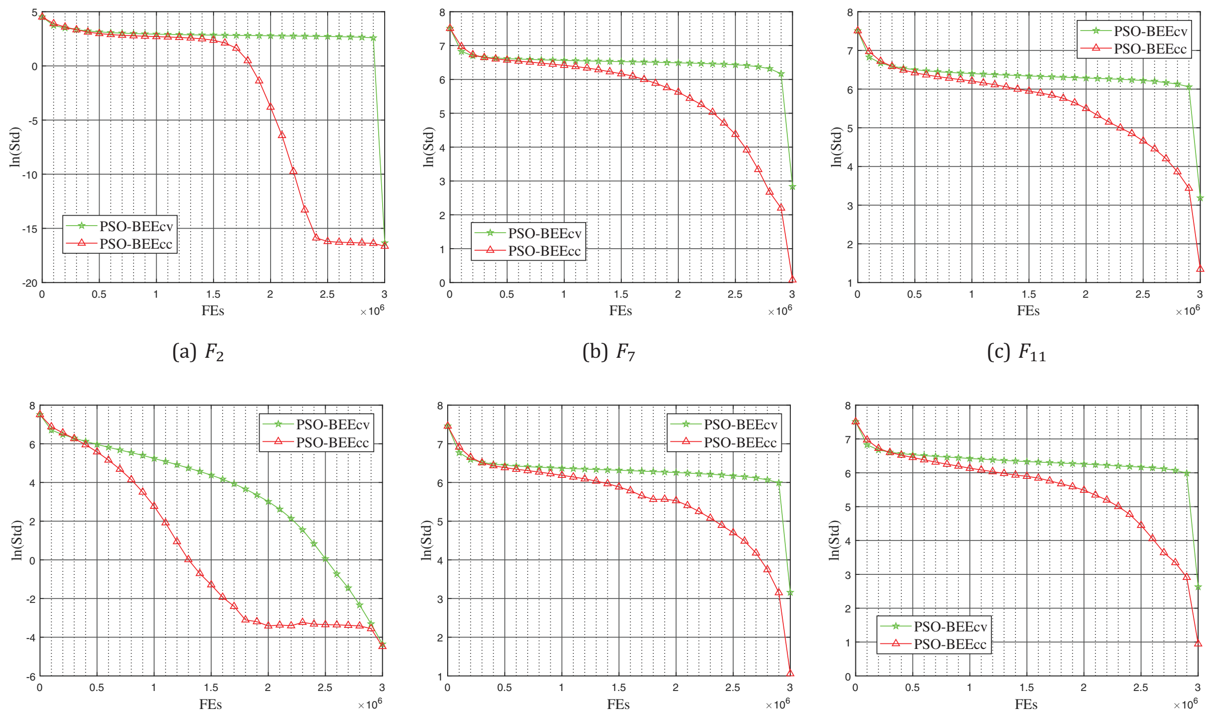
From Fig. 4, first, it can be seen that PSO-BEEcv maintains a higher swarm diversity than PSO-BEEcc in most periods of the evolutionary process, which verifies the effectiveness of the threshold $\boldsymbol{T}$ in controlling the swarm diversity. Second, PSO-BEEcv maintains a high swarm diversity almost throughout the evolutionary process. It is not until nearing the end that the diversity decays rapidly and the swarm begins to converge, which largely prolongs the period for exploration and shortens the stage for exploitation. As a result, PSO-BEEcv is likely to traverse the search space and locate more promising areas than PSO-BEEcc, but it may be difficult to find acceptable solutions in these promising areas because of its poor exploitation ability. That is, PSO-BEEcv could derive the emergence of over-exploration.

Overall, PSO-BEEcc is the one with the best performance on solution quality among the three different versions of PSO-BEE, thus being selected in the comparison with other state-of-the-art algorithms in Sec. 4.2.2. The diversity differences between PSO-BEEcv and PSO-BEEcc verify the feasibility of the threshold $\boldsymbol{T}$ in controlling the swarm diversity.

4.2.2. *Comparison with state-of-the-art algorithms*

The compared algorithms selected are: random contrastive interaction-based PSO (RCI-PSO) [35], adaptive stochastic dominant learning swarm optimizer (SDLSO) [14], adaptive particle swarm optimizer with decoupled exploration and exploitation (APSO-DEE) [33], two-phase learning-based swarm optimizer (TPLSO) [32], level-based learning swarm optimizer (DLLSO) [36], competitive swarm optimizer (CSO) [12], a faster and more accurate differential grouping approach (DECC-DG2) [37], merged differential grouping approach (DECC-MDG) [38], and LSHADE-SPA memetic framework (MLSHADE-SPA) [39].

For these algorithms, RCI-PSO, SDLSO, APSO-DEE, and TPLSO are four competitive algorithms put forward in the past five years, where APSO-DEE is designed based on an exploration and exploitation decoupled learning structure. DLLSO and CSO are two representative swarm optimizers proposed earlier. These PSO variants are proposed by designing novel learning strategies. DECC-DG2 and DECC-MDG are two cooperative coevolutionary framework-based



(a) F_2 (b) F_7 (c) F_{11}

(d) F_{12} (e) F_{13} (f) F_{14}

Fig. 4. Comparisons of diversity characteristics between PSO-BEEcv and PSO-BEEcc obtained on the CEC'2013 large-scale benchmark with 3E+06 fitness evaluations.

algorithms. MLSHADE-SPA is a hybrid algorithm that combines three approaches. MLSHADE-SPA obtains excellent performances in the CEC'2013 benchmark [30] and achieves second place in the CEC'2018 competition on large-scale global optimization.

To ensure a fair comparison, we adopt the optimal parameters of each algorithm that have been adjusted on the same benchmark, rather than a set of identical parameters. Therefore, the parameters between different algorithms vary slightly, which are listed in detail in Table 3. Where NP is the swarm size and ϕ is a balance parameter. Note that this table omits the parameters of DECC-DG2, DECC-MDG, and MLSHADE-SPA, which differ structurally from the proposed algorithm and each adopt some other forms of complex parameters.

Table 3. Main parameters of the proposed and compared PSO variant algorithms on 1000-D optimization.

Algorithm	Main parameters
PSO-BEEcc	$\mathrm{NP}=500,\ \phi=0.4$
RCI-PSO [35]	$\mathrm{NP}=900,\ \phi=0.3$
SDLSO [14]	$\mathrm{NP}=400,\ \phi=0.5-0.2\times\exp(-\frac{f(x_i)-f(x_{b2})+\xi}{f(x_i)-f(x_{b1})+\xi})$
APSO-DEE [33]	$\mathrm{NP}=1000,\ \phi=0.3$
TPLSO [32]	$\mathrm{NP}=600,\ \phi=0.15$
DLLSO [36]	$\mathrm{NP}=500,\ \phi=0.4$
CSO [12]	$\mathrm{NP}=500,\ \phi=0.15$

Tables 4 and 5 exhibit the experimental results of the proposed algorithm and other state-of-the-art approaches obtained on the CEC'2010 benchmark [29] and the CEC'2013 benchmark [30], respectively. The average value, standard deviation, and statistical results are included in the two tables. The symbols, "+", "−", and "=" above the p-value indicate that PSO-BEEcc significantly better than, significantly worse than, and equivalent to compared peer algorithms on associated functions. "$w/t/l$" denotes that PSO-BEEcc obtains significantly better performance on w functions, equivalent performance on t functions, and significantly worse performance on l functions. The average rank of each algorithm over all functions in the two benchmark suites are obtained by utilizing the Friedman test at the significance level of $\alpha = 0.05$.

From Table 4, on the CEC'2010 benchmark, PSO-BEEcc wins first place on 6 out of 20 functions (F_1, F_4, F_7, F_9, F_{14}, and F_{16}) and obtains second place on 10 out of 20 functions (F_3, F_6, F_8, F_{11} to F_{13}, and F_{17} to F_{20}). Besides, PSO-BEEcc has particularly outstanding performance on F_1, F_4, F_7 to F_9, F_{12}, and F_{20}. On these functions, the solutions obtained by PSO-BEEcc have an order or even multiple orders of magnitude improvement compared with those solutions

Table 4. Comparison results between PSO-BEEcc and state-of-the-art algorithms on the 1000-*D* CEC'2010 benchmark with 3E+06 fitness evaluations.

Function	Quality	PSO-BEEcc	RCI-PSO	SDLSO	APSO-DEE	TPLSO	DLLSO	CSO	DECC-DG2	DECC-MDG	MLSHADE-SPA
F_1	Mean	3.26E−27	1.82E−22	5.88E−23	1.08E−20	4.60E−20	2.70E−22	1.95E−18	1.46E+04	1.51E+00	1.74E−22
	Std	3.43E−27	2.21E−23	9.81E−24	1.35E−21	2.93E−20	6.46E−23	8.14E−20	4.74E+04	3.23E+00	3.72E−22
	p-value	—	1.41E−09^{+}	1.41E−09^{+}	1.41E−09^{+}	1.41E−09^{+}	1.41E−09^{+}	1.41E−09^{+}	1.75E−07^{+}	1.41E−09^{+}	1.20E−08^{+}
F_2	Mean	8.29E+02	1.00E+03	9.00E+02	5.81E+02	1.23E+03	9.85E+02	6.34E+02	4.44E+03	4.43E+03	4.98E+01
	Std	3.40E+01	3.81E+01	3.73E+01	3.03E+01	8.21E+01	4.14E+01	2.98E+01	1.59E+02	1.77E+02	7.05E+00
	p-value	—	1.42E−09^{+}	5.56E−07^{+}	1.42E−09^{-}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{-}	1.75E−07^{+}	1.42E−09^{+}	1.21E−08^{-}
F_3	Mean	7.19E−14	8.10E−14	4.38E−14	3.66E−13	1.48E+00	9.01E−14	2.89E−13	1.66E+01	1.67E+01	9.77E−14
	Std	3.88E−15	4.92E−15	3.20E−15	3.06E−14	1.85E−01	5.70E−15	3.26E−15	2.45E−01	3.32E−01	8.51E−15
	p-value	—	1.52E−08^{+}	3.50E−10^{-}	9.21E−10^{+}	9.32E−10^{+}	7.90E−10^{+}	5.85E−10^{+} \eqno {\rm }	9.36E−08^{+}	9.33E−10^{+}	6.89E−09^{+}
F_4	Mean	6.64E+10	9.84E+11	6.14E+11	2.74E+11	4.86E+11	7.64E+11	1.52E+12	4.66E+12	6.87E+11	1.57E+11
	Std	1.00E+10	1.09E+11	1.26E+11	3.72E+10	1.42E+11	1.30E+11	2.15E+11	1.72E+12	2.36E+11	8.27E+10
	p-value	—	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.75E−07^{+}	1.42E−09^{+}	1.67E−05^{+}
F_5	Mean	1.37E+08	1.32E+07	1.05E+07	1.02E+07	2.30E+07	1.34E+07	1.13E+07	1.56E+08	1.35E+08	5.95E+07
	Std	1.31E+08	1.91E+06	2.26E+06	2.12E+06	8.51E+06	2.42E+06	2.62E+06	1.55E+07	1.91E+07	6.93E+06
	p-value	—	1.42E−09^{-}	1.42E−09^{-}	1.42E−09^{-}	6.38E−04^{-}	2.03E−09^{-}	1.42E−09^{-}	5.39E−01$^{=}$	4.73E−01$^{=}$	5.00E−01$^{=}$
F_6	Mean	3.55E−09	3.55E−09	3.55E−09	3.72E−09	2.49E+00	2.96E−01	2.05E−07	1.63E+01	1.62E+01	6.08E−09
	Std	6.84E−15	8.20E−15	3.45E−15	1.83E−11	4.02E−01	4.93E−01	5.25E−09	3.64E−01	3.52E−01	1.66E−09
	p-value	—	5.82E−04^{+}	7.00E−10^{-}	1.16E−09^{+}	1.16E−09^{+}	5.78E−02^{+}	1.16E−09^{+}	1.30E−07^{+}	1.16E−09^{+}	9.41E−09^{+}
F_7	Mean	4.41E−09	1.64E+01	5.29E+00	6.17E−02	2.02E+03	9.99E+00	1.87E+04	1.14E+04	7.70E+01	2.71E−05
	Std	3.48E−09	1.06E+01	4.19E+00	5.71E−02	4.26E+03	1.97E+01	8.24E+03	9.18E+03	2.02E+02	9.12E−05
	p-value	—	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.75E−07^{+}	1.42E−09^{+}	1.21E−08^{+}
F_8	Mean	2.87E+03	2.78E+07	1.90E+07	6.45E+06	1.18E+07	2.86E+07	4.12E+07	2.89E+07	3.19E+05	7.09E−03
	Std	1.25E+03	3.75E+05	2.47E+05	1.51E+06	1.65E+06	1.87E+07	1.49E+07	2.13E+07	1.10E+06	3.09E−02
	p-value	—	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.75E−07^{+}	4.54E−07^{+}	1.21E−08^{-}
F_9	Mean	8.64E+06	5.68E+07	2.77E+07	1.75E+07	4.49E+07	3.83E+07	4.85E+07	5.98E+07	4.51E+07	1.59E+07
	Std	6.38E+05	3.86E+06	2.58E+06	1.17E+06	5.26E+06	3.26E+06	3.33E+06	9.11E+06	5.58E+06	3.04E+06
	p-value	—	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.75E−07^{+}	1.42E−09^{+}	1.21E−08^{+}
F_{10}	Mean	9.44E+02	1.02E+03	8.33E+02	5.86E+02	3.24E+03	9.12E+02	4.66E+02	4.54E+03	4.32E+03	5.02E+03
	Std	3.87E+01	3.69E+01	2.32E+01	2.29E+01	3.27E+03	5.00E+01	1.99E+01	2.01E+02	1.08E+02	1.81E+02
	p-value	—	3.70E−07^{+}	1.42E−09^{-}	1.42E−09^{-}	1.42E−09^{+}	1.99E−02^{-}	1.42E−09^{-}	1.75E−07^{+}	1.42E−09^{+}	1.21E−08^{+}
F_{11}	Mean	1.77E−13	1.87E−13	1.36E−13	9.24E−13	2.40E+04	1.44E+04	3.35E−12	1.03E+01	1.01E+01	9.21E+01
	Std	9.31E−15	9.42E−15	3.96E−15	1.95E−13	1.01E+00	5.35E+00	6.49E−14	1.17E+00	7.53E−01	1.44E+01
	p-value	—	5.67E−04^{+}	1.02E−09^{-}	1.35E−09^{+}	1.36E−09^{+}	1.36E−09^{+}	1.35E−09^{+}	1.64E−07^{+}	1.36E−09^{+}	1.15E−08^{+}
F_{12}	Mean	5.99E+02	4.11E+04	1.44E+04	1.55E+04	3.17E+03	1.42E+03	4.92E+04	2.44E+03	1.40E+03	4.39E+01
	Std	8.57E+01	2.46E+03	1.17E+03	1.16E+03	3.17E+03	1.42E+03	3.07E+03	3.71E+02	2.84E+02	8.76E+00
	p-value	—	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.75E−07^{+}	1.42E−09^{+}	1.21E−08^{-}
F_{13}	Mean	4.77E+02	7.71E+02	9.11E+02	6.32E+02	9.80E+02	1.03E+03	1.35E+03	1.14E+06	8.71E+02	7.29E+01
	Std	8.69E+01	1.82E+02	3.77E+02	1.37E+02	2.49E+02	3.03E+02	4.92E+02	5.90E+05	4.76E+02	4.84E+01
	p-value	—	1.31E−07^{+}	6.57E−09^{+}	7.55E−05^{+}	1.42E−09^{+}	1.80E−09^{+}	1.80E−09^{+}	1.75E−07^{+}	6.80E−07^{+}	1.21E−08^{-}
F_{14}	Mean	2.91E+07	2.28E+08	9.12E+07	6.07E+07	1.48E+08	1.30E+08	1.76E+08	3.40E+08	3.38E+08	5.66E+07
	Std	1.48E+06	9.16E+06	4.08E+06	3.70E+06	1.02E+07	5.86E+06	9.61E+06	1.92E+07	2.32E+07	5.01E+06
	p-value	—	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.75E−07^{+}	1.42E−09^{+}	1.21E−08^{+}
F_{15}	Mean	3.18E+03	1.03E+03	9.65E+02	6.43E+02	1.03E+04	1.58E+03	9.63E+03	5.87E+03	5.85E+03	3.83E+03
	Std	2.83E+03	4.15E+01	1.81E+03	2.81E+01	4.48E+01	2.51E+03	6.59E+01	7.56E+01	8.83E+01	8.17E+01
	p-value	—	1.42E−09^{-}	1.84E−08^{-}	1.42E−09^{-}	1.42E−09^{+}	2.21E−07^{-}	4.26E−06^{+}	7.28E−05^{+}	4.26E−06^{+}	6.38E−04^{+}
F_{16}	Mean	1.73E−13	3.52E−02	1.58E−01	6.86E−13	1.86E+01	3.47E+00	5.62E−12	7.29E−13	2.46E−13	2.54E+02
	Std	1.20E−14	1.76E−01	3.71E−01	5.47E−14	6.83E+00	2.39E+00	1.06E−13	4.69E−14	7.74E−15	1.85E+01
	p-value	—	9.06E−09^{+}	6.76E−07^{+}	1.37E−09^{+}	1.38E−09^{+}	1.38E−09^{+}	1.37E−09^{+}	1.67E−07^{+}	1.30E−09^{+}	1.16E−08^{+}
F_{17}	Mean	1.41E+04	1.73E+05	1.08E+05	9.59E+04	1.43E+05	7.89E+04	2.33E+05	3.94E+04	3.95E+04	6.18E+02
	Std	1.70E+03	8.26E+03	4.69E+03	5.69E+03	1.27E+04	5.19E+03	1.01E+04	2.07E+03	1.96E+03	7.12E+01
	p-value	—	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.42E−09^{+}	1.75E−07^{+}	1.42E−09^{+}	1.21E−08^{-}
F_{18}	Mean	1.00E+03	1.98E+03	1.84E+03	1.23E+03	2.60E+03	2.56E+03	2.37E+03	7.38E+09	1.18E+03	5.96E+02
	Std	1.09E+02	5.55E+02	5.18E+02	2.29E+02	7.40E+02	9.32E+02	6.31E+02	1.24E+09	1.42E+02	2.49E+02

Table 4. (*Continued*)

Function	Quality	PSO-BEEcc	RCI-PSO	SDLSO	APSO-DEE	TPLSO	DLLSO	CSO	DECC-DG2	DECC-MDG	MLSHADE-SPA
	p-value	—	1.42E−09+	5.21E−09+	1.13E−04+	6.64E+06+	1.72E+06+	1.42E−09+	1.75E−07+	1.04E−04+	2.93E−07−
F_{19}	Mean	1.27E+06	1.34E+06	4.58E+06	1.40E+06	6.64E+06	1.72E+06	5.75E+06	1.70E+06	1.73E+06	4.50E+05
	Std	6.75E+04	5.57E+04	3.30E+05	5.06E+04	9.90E+05	7.88E+04	2.60E+05	9.54E+04	8.56E+04	4.67E+04
	p-value	—	4.08E−03+	1.42E−09+	2.87E−08+	1.42E−09+	1.42E−09+	1.42E−09+	1.75E−07+	1.42E−09+	1.21E−08−
F_{20}	Mean	9.57E+02	1.38E+03	1.35E+03	1.04E+03	1.91E+03	1.71E+03	1.28E+03	3.20E+08	4.17E+03	9.46E+01
	Std	3.40E+01	1.52E+02	1.11E+02	4.63E+01	1.47E+02	1.31E+02	1.23E+02	3.14E+08	2.28E+03	1.22E+02
	p-value	—	1.42E−09+	1.42E−09+	5.03E−07+	1.42E−09+	1.42E−09+	1.42E−09+	1.75E−07+	1.42E−09+	1.21E−08−
w/t/l		—	18/0/2	14/0/6	16/0/4	19/0/1	16/1/3	17/0/3	19/1/0	19/1/0	11/1/8
Friedman ranking		**2.45**	5.65	4.30	3.70	7.50	6.00	6.95	8.40	6.55	3.50

Table 5. Comparison results between PSO-BEEcc and state-of-the-art algorithms on the CEC'2013 large-scale benchmark with 3E+06 fitness evaluations.

Function	Quality	PSO-BEEcc	RCI-PSO	SDLSO	APSO-DEE	TPLSO	DLLSO	CSO	DECC-DG2	DECC-MDG	MLSHADE-SPA
F_1	Mean	4.39E−27	2.67E−22	8.76E−23	4.30E−20	1.07E−17	4.00E−22	1.78E−18	1.83E+03	2.35E+01	3.79E−22
	Std	6.87E−27	6.21E−23	1.43E−23	1.16E−20	1.34E−17	9.35E−23	9.42E−20	7.38E+03	4.76E+01	8.18E−22
	p-value	—	7.35E−10+	4.75E−10+	3.76E−06+	7.35E−10+	7.35E−10+	4.75E−10+	7.35E−10+	4.75E−10+	7.35E−10+
F_2	Mean	8.01E+02	9.76E+02	9.95E+02	6.32E+02	1.30E+03	1.16E+03	7.03E+02	1.25E+04	1.26E+04	8.32E+01
	Std	4.12E+01	5.01E+01	6.30E+01	3.40E+01	6.49E+01	6.06E+01	3.25E+01	6.60E+02	5.65E+02	1.31E+01
	p-value	—	8.26E−10+	4.75E−10+	3.76E−06−	7.35E−10+	7.35E−10+	3.89E−09−	7.35E−10+	4.75E−10+	7.35E−10−
F_3	Mean	2.16E+01	2.16E+01	2.16E+01	2.16E+01	2.16E+01	2.16E+01	2.16E+01	2.14E+01	2.14E+01	9.71E−14
	Std	5.56E−03	5.87E−03	4.01E−03	3.54E−03	8.90E−03	5.67E−03	7.34E−03	1.89E−02	1.14E−02	7.75E−15
	p-value	—	6.14E−01=	7.42E−01=	4.97E−01=	7.56E−04−	4.04E−01=	6.52E−02=	7.35E−10−	4.75E−10−	6.60E−10−
F_4	Mean	4.91E+08	6.66E+09	5.58E+09	2.18E+09	4.89E+09	6.57E+09	1.10E+10	1.25E+11	4.38E+10	8.25E+08
	Std	8.04E+07	7.10E+08	1.00E+09	5.66E+08	1.37E+09	2.27E+09	1.16E+09	5.41E+10	1.97E+10	3.50E+08
	p-value	—	7.35E−10+	4.75E−10+	3.76E−06+	7.35E−10+	7.35E−10+	4.75E−10+	7.35E−10+	4.75E−10+	1.50E−04+
F_5	Mean	7.39E+05	7.47E+05	7.30E+05	7.28E+05	6.35E+05	6.87E+05	7.69E+05	7.65E+06	5.02E+06	1.81E+06
	Std	1.21E+05	9.85E+04	1.37E+05	1.06E+05	1.13E+05	1.15E+05	1.13E+05	4.05E+05	4.38E+05	2.41E+05
	p-value	—	5.15E−01=	8.80E−01=	9.60E−01=	1.98E−03−	2.22E−01=	3.59E−01=	7.35E−10+	4.75E−10+	7.35E−10+
F_6	Mean	1.06E+06	1.06E+06	1.06E+06	1.06E+06	1.06E+06	1.06E+06	1.06E+06	1.06E+06	1.06E+06	1.80E+03
	Std	1.43E+03	1.32E+03	1.88E+03	1.10E+03	1.13E+03	1.02E+03	1.15E+03	1.19E+03	1.35E+03	2.59E+03
	p-value	—	3.26E−01=	2.73E−01=	9.60E−01=	3.00E−01=	8.76E−01=	5.75E−01=	1.83E−02+	7.34E−04+	7.35E−10−
F_7	Mean	4.54E+04	4.09E+06	1.59E+06	5.13E+05	1.27E+06	1.78E+06	7.11E+06	6.67E+07	5.43E+07	6.42E+04
	Std	4.21E+04	1.37E+06	8.69E+05	1.57E+05	4.64E+05	8.61E+05	2.03E+06	2.14E+07	1.87E+07	5.41E+04
	p-value	—	7.35E−10+	4.75E−10+	3.76E−06+	7.35E−10+	7.35E−10+	4.75E−10+	7.35E−10+	4.75E−10+	1.16E−02+
F_8	Mean	1.82E+13	1.33E+14	8.99E+13	4.33E+13	6.63E+13	1.22E+14	3.07E+14	8.24E+15	4.54E+15	1.41E+13
	Std	6.08E+12	2.32E+13	3.84E+13	7.31E+12	2.20E+13	2.97E+13	7.24E+13	2.55E+15	1.41E+15	7.59E+12
	p-value	—	7.35E−10+	4.75E−10+	4.41E−06+	2.30E−09+	7.35E−10+	4.75E−10+	7.35E−10+	4.75E−10+	5.75E−02=
F_9	Mean	4.07E+07	3.99E+07	4.30E+07	4.08E+07	4.36E+07	4.63E+07	4.46E+07	5.95E+08	4.86E+08	1.47E+08
	Std	5.91E+06	5.80E+06	9.50E+06	4.80E+06	8.62E+06	8.98E+06	7.83E+06	3.30E+07	2.96E+07	1.83E+07
	p-value	—	8.04E−01=	4.71E−01=	7.53E−01=	2.75E−01=	3.10E−02+	3.18E−02+	7.35E−10+	4.75E−10+	7.35E−10+
F_{10}	Mean	9.40E+07	9.40E+07	9.41E+07	9.40E+07	9.40E+07	9.41E+07	9.40E+07	9.49E+07	9.45E+07	6.10E+02
	Std	2.38E+05	1.76E+05	1.73E+05	9.14E+04	3.28E+05	1.51E+05	2.28E+05	3.61E+05	2.15E+05	2.56E+02
	p-value	—	8.04E−01=	6.62E−01=	9.87E−01=	4.80E−01=	4.04E−01=	3.78E−01=	2.05E−08+	4.28E−08+	7.35E−10−
F_{11}	Mean	9.06E+06	1.96E+08	2.07E+08	7.24E+07	2.16E+08	2.22E+08	3.49E+08	7.19E+09	3.77E+09	4.03E+07
	Std	3.61E+06	3.02E+07	4.82E+07	2.01E+07	1.22E+08	5.26E+07	8.06E+07	6.78E+09	5.33E+09	1.71E+07
	p-value	—	7.35E−10+	4.75E−10+	3.76E−06+	7.35E−10+	7.35E−10+	4.75E−10+	7.35E−10+	4.75E−10+	9.26E−10+
F_{12}	Mean	1.03E+03	1.49E+03	1.42E+03	1.18E+03	2.01E+03	1.81E+03	1.36E+03	7.22E+04	9.45E+04	1.09E+02
	Std	4.36E+01	1.05E+02	1.05E+02	6.61E+01	1.90E+02	1.75E+02	1.02E+02	1.21E+05	4.51E+05	9.14E+01
	p-value	—	7.35E−10+	4.75E−10+	8.25E−06+	7.35E−10+	7.35E−10+	4.75E−10+	7.35E−10+	4.75E−10+	7.35E−10−
F_{13}	Mean	4.54E+06	3.13E+08	4.38E+08	6.35E+07	2.94E+08	3.26E+08	7.01E+08	2.72E+09	1.24E+09	7.86E+07
	Std	2.79E+06	1.11E+08	1.58E+08	3.15E+07	1.93E+08	1.87E+08	1.42E+08	8.96E+08	3.40E+08	5.32E+07
	p-value	—	7.35E−10+	4.75E−10+	3.76E−06+	7.35E−10+	7.35E−10+	4.75E−10+	7.35E−10+	4.75E−10+	9.26E−10+

Table 5. (*Continued*)

Function	Quality	PSO-BEEcc	RCI-PSO	SDLSO	APSO-DEE	TPLSO	DLLSO	CSO	DECC-DG2	DECC-MDG	MLSHADE-SPA
F_{14}	Mean	1.09E+07	2.26E+08	3.61E+08	3.79E+07	1.45E+08	1.15E+08	4.96E+09	6.56E+10	3.70E+09	1.42E+07
	Std	1.29E+06	2.87E+08	3.14E+08	4.40E+06	5.50E+07	9.19E+07	3.44E+09	2.56E+10	1.62E+09	2.33E+06
	p-value	—	7.35E−10$^+$	4.75E−10$^+$	3.76E−06$^+$	7.35E−10$^+$	7.35E−10$^+$	4.75E−10$^+$	7.35E−10$^+$	4.75E−10$^+$	1.90E−06$^+$
F_{15}	Mean	3.36E+06	2.82E+06	1.15E+07	3.07E+06	5.06E+07	4.37E+06	1.67E+07	3.67E+07	1.04E+07	2.71E+07
	Std	4.70E+05	1.81E+05	7.03E+05	3.19E+05	7.00E+06	3.64E+05	1.14E+06	6.95E+06	2.19E+06	8.60E+06
	p-value	—	1.01E−05$^-$	4.75E−10$^+$	5.66E−02$^=$	7.35E−10$^+$	5.26E−08$^+$	4.75E−10$^+$	7.35E−10$^+$	4.75E−10$^+$	7.35E−10$^+$
w/t/l		—	9/5/1	10/5/0	8/6/1	10/3/2	11/4/0	10/4/1	14/0/1	14/0/1	9/1/5
Friedman ranking		**2.73**	5.00	5.43	3.53	5.60	5.83	6.67	9.03	8.17	3.00

obtained by other PSO variants. However, PSO-BEEcc exhibits poor performance on F_5 and F_{15}. More rigorously, PSO-BEEcc behaves very erratically on the two functions; the standard deviation values obtained are even close to the corresponding mean values. Both the two functions are *Rastrigin* function, a typical nonlinear multimodal function characterized by a large number of local extreme points. If the swarm in PSO-BEEcc fails to acquire enough diversity at the early stage (which may be caused by updated particles selecting exemplars that are closely distributed in the objective space or decision space, or even selecting the exactly same exemplars, etc.) to let individuals be distributed extensively in the search space, it will be challenging for the swarm to leave current locally optimal areas for better ones due to the shrinking size of ***CEG***.

The statistical results demonstrate that PSO-BEEcc significantly outperforms RCI-PSO, SDLSO, APSO-DEE, TPLSO, DLLSO, and CSO 18, 14, 16, 19, 16, and 17 times, respectively. As for CC framework-based algorithms, PSO-BEEcc significantly outperforms both DECC-DG2 and DECC-MDG 19 times. For the hybrid algorithm MLSHADE-SPA, PSO-BEEcc is significantly better than it 11 times and significantly worse than it 8 times.

From Table 5, on the CEC'2013 benchmark, PSO-BEEcc wins first place on 6 out of 15 functions (F_1, F_4, F_7, F_{11}, F_{13}, and F_{14}) and obtains second place on 3 functions (F_8, F_9, and F_{12}). Moreover, PSO-BEEcc has particularly outstanding performance on F_1, F_4, F_7, F_{11}, and F_{13}. The solutions obtained by PSO-BEEcc on these functions have an order or even multiple orders of magnitude improvement compared with those solutions obtained by other PSO variants on these functions. However, as for F_3, F_5, F_6, and F_{10}, there are basically no significant differences between the solutions obtained by PSO-BEEcc and other PSO variants. The statistical results show that PSO-BEEcc significantly outperforms RCI-PSO, SDLSO, APSO-DEE, TPLSO, DLLSO, and CSO 9, 10, 8, 10, 11, and 10 times, respectively. For the two CC framework-based algorithms, PSO-BEEcc significantly outperforms each of them 14 times. Last, PSO-BEEcc significantly outperforms MLSHADE-SPA 9 times and significantly being outperformed 5 times. The Friedman test results at the bottom of the two Tables also demonstrate the strong competitiveness or even superior performance of the proposed algorithm. On both the two benchmark suites, in specific, PSO-BEEcc gets the highest rank, MLSHADE-SPA and APSO-DEE obtain the second and third rank, respectively.

Table 6 summarizes the statistical comparison results between PSO-BEEcc and compared algorithms on the CEC'2010 benchmark and the CEC'2013 benchmark. First, for a total of 35 test functions combined from the two benchmark suites, PSO-BEEcc significantly outperforms RCI-PSO, SDLSO, APSO-DEE, TPLSO, DLLSO, and CSO 27, 24, 24, 29, 27, and 27 times; significantly outperforms both DECC-DG2 and DECC-MDG 33 times; significantly outperforms MLSHADE-SPA 20 times; thus achieves the highest Friedman ranking. Second, PSO-BEEcc has competitive performance on both unimodal and multimodal as well as separable and nonseparable functions. Among all functions on which the solutions obtained by PSO-BEEcc have orders of magnitude improvement, most of them are unimodal functions. This should be attributed to the strong convergence characteristics and exploitation ability of the swarm in PSO-BEEcc at the later stage. The excellent performance of PSO-BEEcc on unimodal functions does not come at the expense of the performance on multimodal functions, because the solutions obtained by PSO-BEEcc on multimodal functions are still promising, some of which are even the best.

The convergence characteristics of PSO-BEEcc and compared algorithms on total 35 functions in the CEC'2010 benchmark (F_1 to F_{20}) and the CEC'2013 benchmark (F_1 to F_{15}) are shown in Figs. S1 and S2 in the supplementary material, respectively. From these figures, it can be seen that PSO-BEEcc converges more slowly at the early stages compared with other algorithms. At this stage, the swarm slows down the convergence speed and maintains a higher swarm diversity, which encourages the swarm to focus on exploration. From the middle or later stages, the swarm begins to accelerate the convergence procedure and may converge to a better solution at the end of evolution. That is,

Table 6. Summarized statistical comparisons between PSO-BEEcc and state-of-the-art algorithms on the 1000-*D* CEC’2010 benchmark and CEC’2013 large-scale benchmark.

Benchmark	Problem property	Index	PSO-BEEcc	RCI-PSO	SDLSO	APSO-DEE	TPLSO	DLLSO	CSO	DG2	MDG	MLSHADE-SPA
CEC’2010	Fully Separable Unimodal	*w/t/l*	—	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0
	Fully Separable Multimodal	*w/t/l*	—	2/0/0	1/0/1	1/0/1	2/0/0	2/0/0	1/0/1	2/0/0	2/0/0	1/0/1
	Partially Separable Unimodal	*w/t/l*	—	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	4/0/2
	Partially Separable Multimodal	*w/t/l*	—	7/0/2	4/0/5	6/0/3	8/0/1	5/1/3	7/0/2	8/1/0	8/1/0	5/1/3
	Fully Nonseparable Unimodal	*w/t/l*	—	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	0/0/1
	Fully Nonseparable Multimodal	*w/t/l*	—	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	0/0/1
	Overall	*w/t/l*	—	18/0/2	14/0/6	16/0/4	19/0/1	16/1/3	17/0/3	19/1/0	19/1/0	11/1/8
	Overall	*rank*	**2.45**	5.65	4.30	3.70	7.50	6.00	6.95	8.40	6.55	3.50
CEC’2013	Fully Separable Unimodal	*w/t/l*	—	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0
	Fully Separable Multimodal	*w/t/l*	—	1/1/0	1/1/0	0/1/1	1/0/1	1/1/0	0/1/1	1/0/1	1/0/1	0/0/2
	Partially Separable Unimodal	*w/t/l*	—	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	2/1/0
	Partially Separable Multimodal	*w/t/l*	—	1/4/0	1/4/0	1/4/0	1/3/1	2/3/0	2/3/0	5/0/0	5/0/0	3/0/2
	Overlapping Unimodal	*w/t/l*	—	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0
	Overlapping Multimodal	*w/t/l*	—	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	0/0/1
	Fully Nonseparable Unimodal	*w/t/l*	—	0/0/1	1/0/0	0/1/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0
	Overall	*w/t/l*	—	9/5/1	10/5/0	8/6/1	10/3/2	11/4/0	10/4/1	14/0/1	14/0/1	9/1/5
	Overall	*rank*	**2.73**	5.00	5.43	3.53	5.60	5.83	6.67	9.03	8.17	3.00
Overall of the CEC’2010 and CEC’2013 benchmark		*w/t/l*	—	27/5/3	24/5/6	24/6/5	29/3/3	27/5/3	27/4/4	33/1/1	33/1/1	20/2/13
		rank	**2.57**	5.37	4.79	3.63	6.69	5.93	6.83	8.67	7.24	3.29

PSO-BEEcc sacrifices convergence speed in exchange for swarm diversity at the early stage so as to carry out biased exploration behaviors. As a compensation, PSO-BEEcc accelerates the convergence speed for efficient exploitation at the expense of the swarm diversity at the later stage, trying to refine better solutions in promising local areas. These developed strategies take full advantage of both exploration and exploitation, which is the direct reason for the superior performance of PSO-BEEcc. The convergence characteristics of PSO-BEEcc are very ideal and in line with design expectations.

In summary, the proposed PSO-BEEcc has competitive or even superior performance in regard to solution quality and convergence characteristics in comparison with state-of-the-art algorithms. The convergence curves reflect the effectiveness of strategies designed in PSO-BEEcc.

4.2.3. *Parameter sensitivity*

In this section, we try to investigate the effects and sensitivities of each parameter on the performance of PSO-BEEcc independently. There are three parameters in PSO-BEEcc: the swarm size NP, ϕ, and the threshold $\boldsymbol{T}$. Experiments are executed on the CEC’2013 benchmark [30] with 3E+06 fitness evaluations. In our experiments, NP is varied from 300 to 700 with a step of 100, ϕ ranges from 0.1 to 0.5 with a step of 0.1, $\boldsymbol{T}$ is set to $\{0.2, 0.4, 0.6, 0.8\}$. When the effects of each one of the three parameters are being investigated, the rest two parameters are set to their default values.

Table S3 in the supplementary material shows the experimental results obtained by PSO-BEEcc with different settings of NP, where ϕ is set to 0.4 and $\boldsymbol{T}$ is dynamically changed according to (11). From Table S3, first, $\mathrm{NP} = 500$ achieves the highest rank. Second, 700 NP obtains the best average results on 6 out of 15 functions, where these six are all multimodal functions. However, 700 NP only gets the third rank, which is caused by its poor performance on unimodal functions because of its slow convergence procedure and inferior exploitation ability. Third, PSO-BEEcc is not sensitive to NP when $\phi = 0.4$ because the rankings under each setting of NP have little difference.

Table S4 in the supplementary material shows the experimental results obtained by PSO-BEEcc with different settings of ϕ, where NP is set to 500 and $\boldsymbol{T}$ is dynamically changed according to (11). From Table S4, first, $\phi = 0.4$ achieves the highest rank. Second, PSO-BEEcc is also not very sensitive to ϕ. Whereas, to ensure the reliability of solutions obtained, it is recommended that ϕ in PSO-BEEcc should not be less than 0.3 but not greater than 0.5 for 1000-*D* optimization problems.

Table 7 exhibits the Friedman ranking of different NP and ϕ combinations in PSO-BEEcc, where the $\boldsymbol{T}$ is dynamically changed according to (11). From Table 7, first, the combination of $\mathrm{NP} = 500$ and $\phi = 0.4$ achieves the highest rank, thereby becoming the suggested parameter combination. With a deep insight, the higher ranks mainly appear in the fifth column of the Table, which is when $\phi = 0.4$. As for the correlation between NP and ϕ, when NP is relatively small, a relatively large value of ϕ is required to make up for

Table 7. The Friedman ranking on all selected NP and ϕ combinations of PSO-BEEcc on the CEC'2013 large-scale benchmark at the significance level of $\alpha = 0.05$.

NP \ ϕ	0.1	0.2	0.3	0.4	0.5
300	16.80	17.83	15.60	11.07	8.53
400	16.53	13.50	12.77	8.63	10.70
500	16.13	13.30	9.67	**7.33**	11.10
600	17.17	13.73	9.77	9.47	15.73
700	15.97	14.17	11.30	10.67	17.53

the lack of diversity caused by the small swarm size, such as the combination of NP $= 300$ and $\phi = 0.5$, which achieves the second rank. However, when NP is larger, such as NP $= 700$, PSO-BEEcc is hard to obtain reliable solutions no matter how ϕ is chosen. It should be noted that neither NP nor ϕ are parameters that play a major role in regulating the swarm diversity and convergence characteristics in PSO-BEEcc.

Last, our emphasis is placed on the discussion of the effects of the threshold $\boldsymbol{T}$. Table S5 in the supplementary material exhibits the experimental results obtained by PSO-BEE with different settings of $\boldsymbol{T}$, where NP is set to 500 and ϕ is set to 0.4. The experimental results in the last column of Table S5 are obtained when $\boldsymbol{T}$ changes dynamically as a continuous variable according to (11). As comparison groups, the other four discrete values of $\boldsymbol{T}$ are set, namely, 0.2, 0.4, 0.6, and 0.8. From Table S5, first, PSO-BEE with dynamic $\boldsymbol{T}$ obtains the best average results on 10 out of 15 functions and beats all PSO-BEE with fixed $\boldsymbol{T}$ down, thus winning the highest rank. Second, PSO-BEE with a small value of $\boldsymbol{T}$ is more suitable for solving unimodal functions because of its strong convergence characteristics and exploitation ability. Third, PSO-BEE is sensitive to $\boldsymbol{T}$, a parameter that plays a significant role in regulating the swarm convergence and diversity characteristics as well as the exploration and exploitation behaviors.

Figure S3 in the supplementary material exhibits the diversity curves of PSO-BEE with different settings of $\boldsymbol{T}$ on 15 functions (F_1 to F_{15}) in the CEC'2013 large-scale benchmark. From Fig. S3, first, by observing the four diversity curves from $\boldsymbol{T} = 0.2$ to $\boldsymbol{T} = 0.8$, it can be learned that the swarm diversity is positively related to $\boldsymbol{T}$, which means that the larger the value of $\boldsymbol{T}$, the higher the swarm diversity preserved, and the difference is significant. That is the reason why $\boldsymbol{T}$ serves as the key parameter regulating the swarm diversity. Second, when $\boldsymbol{T}$ dynamically changes, the swarm is able to maintain the highest diversity at the early stage and this diversity will decay to the minimum level at the later stage compared with when other fixed $\boldsymbol{T}$ are adopted. This dynamic changing of $\boldsymbol{T}$ endows the swarm with strong exploration ability at the early stage and efficient exploitation ability at the later stage, verifying the effectiveness of strategies designed in PSO-BEE.

4.2.4. *Scalability*

In this section, we investigate the scalability of the proposed algorithm from the perspective of solution quality. PSO-BEEcc and compared algorithms are applied to the CEC'2010 benchmark suite [29] with different dimensionalities for 25 independent runs. All functions in the CEC'2010 benchmark can be scalable to any number of dimensions. Therefore, we regenerate each function in this benchmark with a dimensionality of 500 and 2000. The maximum fitness evaluation Maxfes is set to $3000 \times D$ (D is the dimensionality), namely, 1.5E+06 Maxfes for 500-D functions and 6E+06 Maxfes for 2000-D functions.

Based on the experimental results, the swarm size NP of PSO-BEEcc is set to 400 and 700 for 500-D and 2000-D functions, respectively. For fairness, the most appropriate NP of each compared algorithm for functions with different dimensionalities are adopted, which are listed in RCI-PSO [35] in detail.

Table S6 in the supplementary material exhibits the experimental results obtained on 500-D functions. From Table S6, first, PSO-BEEcc achieves the best average results on 8 out of 20 functions (F_1, F_4, F_6 to F_9, F_{14}, and F_{16}) and obtains second place on 9 out of 20 functions (F_3, F_{10} to F_{13}, F_{17} to F_{20}), which establishes the highest Friedman ranking of PSO-BEEcc. However, the performance of PSO-BEEcc on F_5 and F_{15} is still unsatisfactory despite the lower dimensionality. Second, the statistical results demonstrate that PSO-BEEcc significantly outperforms RCI-PSO, SDLSO, APSO-DEE, TPLSO, DLLSO, and CSO 17, 14, 16, 19, 16, and 18 times, respectively. As for CC framework-based algorithms, PSO-BEEcc significantly outperforms DECC-DG2 and DECC-MDG 18 and 19 times, respectively. For the hybrid algorithm MLSHADE-SPA, PSO-BEEcc is significantly better than it 11 times and significantly worse than it 7 times. Third, except for F_4, F_5, F_7, and F_8, PSO-BEEcc achieves better solutions for other functions of 500 dimensions than for corresponding functions of 1000 dimensions. The experimental results demonstrate the superior performance of PSO-BEEcc for 500-D problems.

Table S7 in the supplementary material exhibits the experimental results obtained on 2000-D functions. From Table S7, first, PSO-BEEcc achieves the best average results on 4 out of 20 functions (F_4, F_9, F_{14}, and F_{18}) and obtains second place on 9 out of 20 functions (F_1, F_6 to F_8, F_{12}, F_{13}, F_{16}, F_{17}, and F_{19}). In the Friedman test, PSO-BEEcc still wins the highest rank. The statistical results demonstrate that the superior performance of PSO-BEEcc holds for 2000-D functions when compared with PSO

Table 8. Summarized statistical comparisons between PSO-BEEcc and state-of-the-art algorithms on the CEC'2010 benchmark with 500, 1000 and 2000 dimensions.

Benchmark	Problem property	Index	PSO-BEEcc	RCI-PSO	SDLSO	APSO-DEE	TPLSO	DLLSO	CSO	DG2	MDG	MLSHADE-SPA
	Fully Separable Unimodal	*w/t/l*	—	0/1/0	0/1/0	1/0/0	1/0/0	0/1/0	1/0/0	1/0/0	1/0/0	1/0/0
	Fully Separable Multimodal	*w/t/l*	—	2/0/0	1/0/1	1/0/1	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	1/0/1
	Partially Separable Unimodal	*w/t/l*	—	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	4/0/2
CEC'2010	Partially Separable Multimodal	*w/t/l*	—	7/0/2	5/1/3	6/0/3	8/0/1	7/0/2	7/0/2	7/2/0	8/1/0	5/2/2
500-*D*	Fully Nonseparable Unimodal	*w/t/l*	—	1/0/0	1/0/0	1/0/0	1/0/0	0/1/0	1/0/0	1/0/0	1/0/0	0/0/1
	Fully Nonseparable Multimodal	*w/t/l*	—	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	0/0/1
	Overall	*w/t/l*	—	17/1/2	14/2/4	16/0/4	19/0/1	16/2/2	18/0/2	18/2/0	19/1/0	11/2/7
	Overall	*rank*	**2.30**	5.68	4.03	3.55	6.33	6.78	6.80	8.20	7.95	3.40
	Fully Separable Unimodal	*w/t/l*	—	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0
	Fully Separable Multimodal	*w/t/l*	—	2/0/0	1/0/1	1/0/1	2/0/0	2/0/0	1/0/1	2/0/0	2/0/0	1/0/1
	Partially Separable Unimodal	*w/t/l*	—	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	4/0/2
CEC'2010	Partially Separable Multimodal	*w/t/l*	—	7/0/2	4/0/5	6/0/3	8/0/1	5/1/3	7/0/2	8/1/0	8/1/0	5/1/3
1000-*D*	Fully Nonseparable Unimodal	*w/t/l*	—	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	0/0/1
	Fully Nonseparable Multimodal	*w/t/l*	—	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	0/0/1
	Overall	*w/t/l*	—	18/0/2	14/0/6	16/0/4	19/0/1	16/1/3	17/0/3	19/1/0	19/1/0	11/1/8
	Overall	*rank*	**2.45**	5.65	4.30	3.70	7.50	6.00	6.95	8.40	6.55	3.50
	Fully Separable Unimodal	*w/t/l*	—	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	0/0/1
	Fully Separable Multimodal	*w/t/l*	—	2/0/0	0/0/2	1/0/1	2/0/0	0/0/2	2/0/0	2/0/0	2/0/0	1/0/1
	Partially Separable Unimodal	*w/t/l*	—	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	5/1/0	3/0/3
CEC'2010	Partially Separable Multimodal	*w/t/l*	—	7/0/2	4/1/4	6/0/3	9/0/0	6/1/2	8/0/1	8/1/0	7/1/1	5/2/2
2000-*D*	Fully Nonseparable Unimodal	*w/t/l*	—	0/0/1	1/0/0	0/1/0	1/0/0	1/0/0	1/0/0	0/1/0	0/1/0	0/0/1
	Fully Nonseparable Multimodal	*w/t/l*	—	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	0/0/1	1/0/0	1/0/0	0/0/1
	Overall	*w/t/l*	—	17/0/3	13/1/6	15/1/4	20/0/0	15/1/4	18/0/2	18/2/0	16/3/1	9/2/9
	Overall	*rank*	**2.78**	5.15	4.78	3.50	7.48	5.58	7.30	8.30	6.70	3.45

variants and CC framework-based algorithms. Second, MLSHADE-SPA shows strong competitiveness on 2000-*D* functions, which achieves the best average results on 9 out of 20 functions and outperforms PSO-BEEcc 9 times. The solutions obtained by MLSHADE-SPA on individual functions with 2000 dimensions are close to or even better than those on the corresponding functions with 1000 dimensions. Third, the higher complexity of objective functions and the exponential growth of the search space due to the increase in dimensionality make it challenging for these tested algorithms, including PSO-BEEcc, to achieve as promising results on most functions with 2000 dimensions as they do when applied to 1000-*D* functions.

Table 8 summarizes the statistical comparison results between PSO-BEEcc and compared algorithms on CEC'2010 benchmark with different dimensionalities. From Table 8, first, PSO-BEEcc achieves optimal performance on benchmark suites with 500, 1000, and 2000 dimensions. Second, the great advantage of PSO-BEEcc in unimodal functions holds for all the different dimensions tested. Third, as for multimodal functions in a higher-dimensional environment, PSO-BEEcc is more effective on partially separable and nonseparable multimodal functions than on fully separable multimodal functions.

Overall, the proposed PSO-BEEcc has good scalability for problems with middle or higher dimensionality in large-scale optimization.

4.3. *Experimental results on high-dimensional feature selection*

4.3.1. *Experimental setup*

In this section, we apply PSO-BEEcc to cope with real-world high-dimensional feature selection and classification problems. Experimental results are compared with those of several state-of-the-art PSO variants that have been proposed in the past five years to verify the effectiveness of strategies designed in PSO-BEEcc.

We first select five datasets from *UCI Machine Learning Repository*. To ensure the high-dimensional optimization environment, the number of features in the selected datasets is no less than 500. The properties of the five selected datasets are listed in Table 9.

In specific, madelon [40] is an artificial dataset, which was part of the NIPS 2003 feature selection challenge; the data used in parkinson's disease classification [41] were gathered from 188 patients with PD (107 men and 81 women) with ages ranging from 33 to 87 (65.1Â $\pm$ 10.9) at

Table 9. Dataset properties.

Dataset	No. of features	No. of instances	No. of classes
madelon [40]	500	2600	2
parkinson [41]	754	756	2
period changer [42]	1177	90	2
toxicity [43]	1203	171	2
warpAR10P [44]	2400	130	10

the Department of Neurology in CerrahpaÅŸa Faculty of Medicine, Istanbul University; period changer [42] includes 90 nontoxic molecules designed for functional domain of a core clock protein. toxicity [43] includes 171 molecules designed for functional domains of a core clock protein, CRY1, responsible for generating circadian rhythm. warpAR10P [44] has the most features among all the selected datasets and is a multi-classification dataset. Experiments on these datasets might advance the development of unmanned systems in the biomedical field.

Since the number of instances in madelon dataset is large, we randomly select 66.7% samples in this dataset as the training set. As for parkinson's disease classification, period changer, toxicity, and warpAR10P datasets, we randomly select 90% samples as the training set.

As arranged in SDLSO [14], we utilize the k-nearest-neighbor (kNN) classifier [45] with $k = 5$ to perform classification tasks, and n-fold cross validation (with $n = 10$) is adopted to reduce the risk of overfitting. The objective function $f(x)$ is defined as the average classification error rate on the training set. In addition, we selected two other criteria to evaluate the performance of the proposed algorithm, which are computational time and the number of selected features. The swarm size NP for all algorithms is set to 200, the maximum fitness evaluation Maxfes is set to $100 \times \mathrm{NP} = 2\mathrm{E}+04$. Each algorithm is employed on the five datasets for 25 independent runs. The key processes of the experiments in this section are elaborated as follows:

(1) Input the dataset and divide it into a training set and a test set according to the given ratio.
(2) Randomly initialize the swarm. For each individual in the swarm, use Algorithm 1 to determine the feature subset selected and remove the data corresponding to the unselected features on the training set.
(3) Utilize the kNN classifier to calculate the fitness of each particle. The swarm iterates to optimize the feature selection solutions within the given fitness evaluations.
(4) Output the final average classification error rate on the training set, the number of selected features, and the computational time.

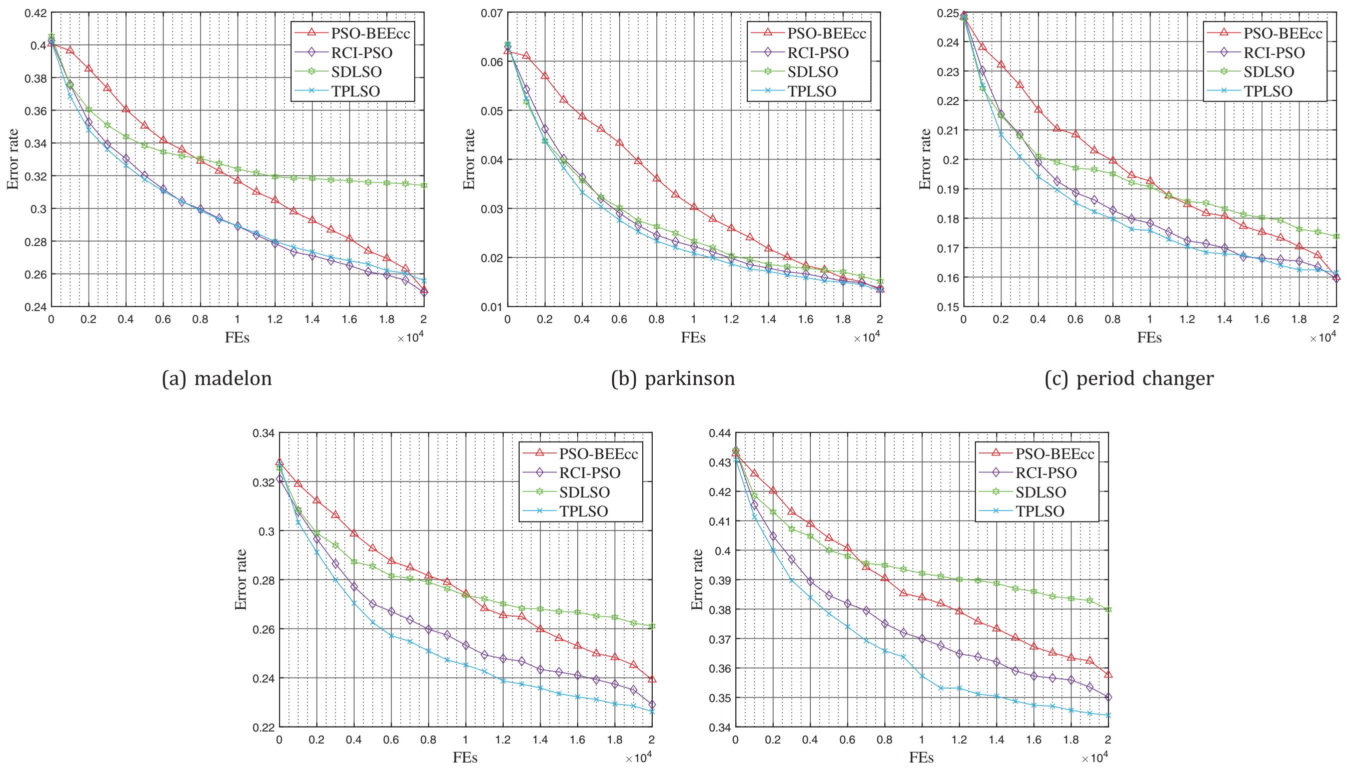
4.3.2. *Classification error rate*

Classification error rate is one of the most important criteria to measure the performance of a classifier. It shows the proportion of misclassified instances in a dataset's total examples. The average classification error rate of different algorithms on the five datasets are exhibited in Table 10.

From Table 10, first, after feature selection by PSO-BEEcc, the average accuracy of kNN on the parkinson dataset improves by about 8.33% compared to directly using kNN for classification tasks. On the madelon, period changer, toxicity, and warpAR10P datasets, the improved average accuracy is about 19%, 16.5%, 15.4%, and 13.5%,

Table 10. Average classification error rate of tested algorithms on different datasets.

Dataset	Quality	PSO-BEEcc-kNN	RCIPSO-kNN	SDLSO-kNN	TPLSO-kNN	kNN
1. madelon	Mean	2.500E−01	2.486E−01	3.140E−01	2.557E−01	4.401E−01
	Std	1.177E−02	1.173E−02	1.294E−02	6.620E−03	1.389E−02
	p-value	—	7.050E−01$^{=}$	1.408E−09^{+}	4.510E−02^{+}	1.407E−09^{+}
2. parkinson	Mean	1.341E−02	1.371E−02	1.512E−02	1.324E−02	9.671E−02
	Std	3.351E−03	3.519E−03	3.665E−03	3.914E−03	6.512E−03
	p-value	—	4.883E−02^{+}	1.684E−02^{+}	8.524E−01$^{=}$	1.307E−09^{+}
3. period changer	Mean	1.600E−01	1.595E−01	1.738E−01	1.615E−01	3.254E−01
	Std	1.902E−02	1.706E−02	1.883E−02	1.917E−02	4.108E−02
	p-value	—	9.762E−01$^{=}$	1.256E−02^{+}	6.614E−01$^{=}$	1.242E−09^{+}
4. toxicity	Mean	2.392E−01	2.291E−01	2.610E−01	2.262E−01	3.930E−01
	Std	1.745E−02	2.560E−02	1.697E−02	1.981E−02	2.704E−02
	p-value	—	1.243E−01$^{=}$	1.607E−04^{+}	2.535E−02^{-}	1.308E−09^{+}
5. wrapAR10P	Mean	3.576E−01	3.501E−01	3.798E−01	3.439E−01	4.923E−01
	Std	2.831E−02	2.617E−02	2.624E−02	3.172E−02	3.986E−02
	p-value	—	2.615E−01$^{=}$	1.451E−02^{+}	1.158E−02^{-}	1.365E−02^{+}
w/t/l		—	1/4/0	5/0/0	1/2/2	5/0/0



(a) madelon (b) parkinson (c) period changer

(d) toxicity (e) warpAR10P

Fig. 5. Comparisons of the trend of classification error rate between PSO-BEEcc and other algorithms obtained on different datasets with 2E+04 fitness evaluations.

respectively. Since warpAR10P contains more features and is a multi-classification dataset, it is quite challenging to deal with. After feature selection, there is still a classification error rate of approximately 35%. Second, feature selection by the PSO variant algorithms significantly reduces the standard deviation of classification error rate. Third, when compared to other PSO variants, PSO-BEEcc significantly outperforms RCI-PSO on the parkinson dataset. On the other four datasets, there is no significant difference between the results obtained by the two algorithms. As for SDLSO, PSO-BEEcc achieves significantly better performance than SDLSO on all selected datasets. In comparison to TPLSO, it outperforms PSO-BEEcc and ties with PSO-BEEcc twice each. In summary, the performance of the proposed algorithm on high-dimensional feature selection is very competitive when compared with state-of-the-art approaches.

The convergence characteristics of tested algorithms on the five datasets are exhibited in Fig. 5, where the swarm convergence is represented by the error rate of the best particle in each generation. From Fig. 5, at the early stage, when PSO-BEEcc is committed to exploring new solutions for feature selection, the error rate decreases more slowly than those of other algorithms. While at the later stage, the error rate of PSO-BEEcc decreases rapidly due to its strong exploitation ability. Experimental results demonstrate the effectiveness of strategies designed in PSO-BEEcc.

4.3.3. *Number of selected features*

A good algorithm should be able to choose as few features as possible while maintaining the classifier's accuracy, which will cut down on the computational resources needed to train the classifier. In this section, we investigate the number of features selected by different algorithms. Experimental results are shown in Table 11.

From Table 11, first, compared to RCI-PSO and TPLSO, PSO-BEEcc selected fewer features on all datasets. Especially on the madelon dataset, the number of features selected by PSO-BEEcc is significantly fewer than those of other algorithms. As for SDLSO, it selected fewer features than that of PSO-BEEcc on the parikinson, period changer, and warpAR10P dataset. However, according to the results shown in Table 10, the *k*NN classifier performs poorly on the feature subset it selects.

The box plot of the number of features selected by different algorithms are exhibited in Fig. 6. We mainly focus on

Table 11. Number of selected features by tested algorithms on different datasets.

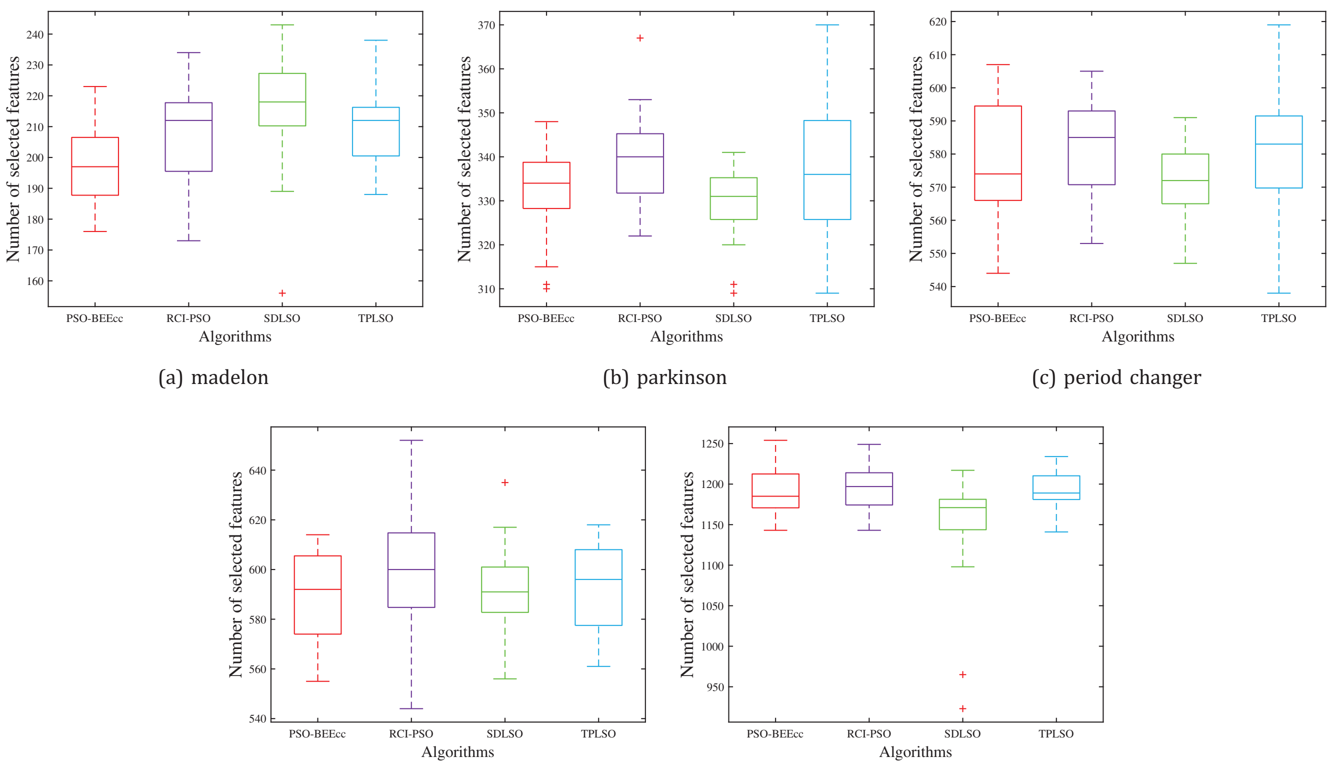
Dataset	Quality	PSO-BEEcc-*k*NN	RCIPSO-*k*NN	SDLSO-*k*NN	TPLSO-*k*NN
1. madelon	Mean	198.88	207.68	216.84	209.88
	Std	13.42	14.41	18.94	12.70
2. parkinson	Mean	332.28	339.36	329.64	338.12
	Std	11.06	10.51	8.17	15.85
3. period changer	Mean	578.64	581.16	571.36	582.80
	Std	16.30	14.43	11.61	19.77
4. toxicity	Mean	589.00	596.88	592.32	594.04
	Std	17.11	24.82	16.41	16.32
5. warpAR10P	Mean	1191.52	1196.48	1149.24	1192.00
	Std	27.06	27.75	67.46	21.42

the median and 25% quantile (Q1). The former reflects the center position of the data, and the latter means that 25% of the data is less than or equal to this value. It can be seen from Fig. 6 that the median of PSO-BEEcc is lower than that of RCI-PSO and TPLSO. As for the Q1, PSO-BEEcc is lower than RCI-PSO on all datasets and is higher than TPLSO only on the parkinson dataset. For SDLSO, although it tends to select fewer features, it cannot guarantee classification accuracy. At the same time, the results of SDLSO contain more outliers, values outside the interquartile range by more than 1.5 times.

4.3.4. *Computational time*

In this section, we investigate the computational time of different algorithms. To ensure a fair comparison, all tested algorithms are vectorized in MATLAB and run on the same hardware environment. Experimental results are shown in Table 12.

From Table 12, first, the increase in the number of instances will raise the computation time more significantly than that in the number of features. The reason is



(a) madelon (b) parkinson (c) period changer

(d) toxicity (e) warpAR10P

Fig. 6. Box plot of the number of features selected by tested algorithms on different datasets.

Table 12. Computational time of tested algorithms on different datasets. The unit of measurement is seconds (s).

Dataset	Quality	PSO-BEEcc-*k*NN	RCIPSO-*k*NN	SDLSO-*k*NN	TPLSO-*k*NN
1. madelon	Mean	53362.57	54273.56	52558.89	53755.35
	Std	5261.12	5997.56	7654.64	3877.35
2. parkinson	Mean	8021.14	8126.07	7987.61	7914.35
	Std	1024.43	809.41	889.66	792.72
3. period changer	Mean	587.16	586.59	579.85	580.30
	Std	34.20	35.11	33.99	34.89
4. toxicity	Mean	1310.51	1320.43	1290.17	1297.93
	Std	103.01	106.02	94.09	102.19
5. warpAR10P	Mean	1386.21	1388.27	1338.09	1372.10
	Std	13.28	23.13	13.80	22.54

as follows: The use of vectorized programming allows for the simultaneous updating of all particle dimensions. This operation is performed in parallel. While the classification of each instance by the *k*NN classifier can only be conduct serially. Second, according to the results in Table 12, unfortunately, PSO-BEEcc tends to consume relatively more computational time than other algorithms.

In summary, the proposed algorithm PSO-BEEcc is able to select fewer features while ensuring the competitive accuracy of the classifier. Since utilizing heuristic algorithms for feature selection consumes a lot of extra computational time, PSO-BEEcc may not be suitable for solving applications that have high requirements for rapid response or real-time performance. For instance, path planning tasks involve autonomous navigation and obstacle avoidance.

5. Conclusion and Future Work

In this paper, a general swarm dynamic threshold segmentation strategy has been proposed, which aims at taking full advantage of exploration and exploitation and is parameter-free. Combined with the exemplar random selection mechanism, a novel particle swarm optimizer with biased exploration and exploitation (PSO-BEE) has been proposed for large-scale global optimization. Three PSO-BEE variants have been subsequently proposed based on different changing trends of the threshold $\boldsymbol{T}$. The three PSO-BEE variants each have distinct preferences for exploration and exploitation. The one with the best performance (PSO-BEEcc) has been selected to be compared with several state-of-the-art approaches. PSO-BEEcc has achieved the highest Friedman ranking on benchmarks with 500, 1000, and 2000 dimensions. PSO-BEEcc has also been employed to deal with high-dimensional feature selection tasks that can be widely found in unmanned systems. Experimental results demonstrated that PSO-BEEcc is able to select fewer features while ensuring the competitive accuracy of the classifier.

However, PSO-BEEcc is found to be time-consuming; it may not be suitable for solving applications that have high requirements for rapid response or real-time performance. Our future work is to design adaptive adjustment strategies for $\boldsymbol{T}$ based on swarm diversity measurement to further improve the performance of PSO-BEE on high-dimensional global optimization and feature selection.

Supplementary Material

The supplementary material is available at https://www.worldscientific.com/doi/suppl/10.1142/S2301385025500840.

ORCID

Minchong Chen https://orcid.org/0009-0006-5678-4651
Hong Li https://orcid.org/0009-0007-7388-618X
Jiwei Tu https://orcid.org/0009-0000-5061-4295
Bin He https://orcid.org/0000-0003-3193-6269
Zhen Yin https://orcid.org/0000-0003-4068-5422
Xuejing Hou https://orcid.org/0009-0003-7705-9908
Qi Yu https://orcid.org/0009-0001-3929-6633

References

[1] F. S. Gharehchopogh and A. A. Khargoush, A chaotic-based interactive autodidactic school algorithm for data clustering problems and its application on COVID-19 disease detection, *Symmetry* **15**(4) (2023) 894.

[2] M. Ayar, A. Isazadeh, F. S. Gharehchopogh and M. Seyedi, Chaotic-based divide-and-conquer feature selection method and its application in cardiac arrhythmia classification, *J. Supercomput.* **78** (2022) 5856–5882.

[3] F. S. Gharehchopogh and T. Ibrikci, An improved african vultures optimization algorithm using different fitness functions for multi-level thresholding image segmentation, *Multimed. Tools Appl.* **83**(6) (2024) 16929–16975.

[4] L. Dai, Y. Zhu, G. Luo, C. He and H. Lin, A real-time visual tracking approach using sparse autoencoder and extreme learning machine, *Unmanned Syst.* **3**(04) (2015) 267–275.

[5] R. Li, H. Shen, L. Wang, C. Liu and X. Yi, A visual feature mismatch detection algorithm for optical flow-based visual odometry, *Unmanned Syst.* (2024).

[6] H. Ali, G. Xiong, M. H. Haider, T. S. Tamir, X. Dong and Z. Shen, Feature selection-based decision model for UAV path planning on rough terrains, *Expert Syst. Appl.* **232** (2023) 120713.

[7] R. Kohavi and G. H. John, Wrappers for feature subset selection, *Artif. Intell.* **97**(1–2) (1997) 273–324.

[8] S. M. Meerkov and M. T. Ravichandran, Combating curse of dimensionality in resilient monitoring systems: Conditions for lossless decomposition, *IEEE Trans. Cybern.* **47**(5) (2016) 1263–1272.

[9] F. Caraffini, F. Neri and G. Iacca, Large scale problems in practice: The effect of dimensionality on the interaction among variables, in *Applications of Evolutionary Computation: 20th Eur. Conf., EvoApplications 2017, Amsterdam, The Netherlands, April 19-21, 2017, Proceedings, Part I 20* (Springer, 2017), pp. 636–652.

[10] J. Kennedy and R. Eberhart, Particle swarm optimization, in *Proc. ICNN'95-Int. Conf. Neural Networks*, Vol. 4 (IEEE, 1995), pp. 1942–1948.

[11] B. Xue, M. Zhang, W. N. Browne and X. Yao, A survey on evolutionary computation approaches to feature selection, *IEEE Trans. Evol. Comput.* **20**(4) (2015) 606–626.

[12] R. Cheng and Y. Jin, A competitive swarm optimizer for large scale optimization, *IEEE Trans. Cybern.* **45**(2) (2014) 191–204.

[13] S. Gu, R. Cheng and Y. Jin, Feature selection for high-dimensional classification using a competitive swarm optimizer, *Soft Comput.* **22** (2018) 811–822.

[14] Q. Yang, W.-N. Chen, T. Gu, H. Jin, W. Mao and J. Zhang, An adaptive stochastic dominant learning swarm optimizer for high-dimensional optimization, *IEEE Trans. Cybern.* **52**(3) (2020) 1960–1976.

[15] J.-Q. Yang, Q.-T. Yang, K.-J. Du, C.-H. Chen, H. Wang, S.-W. Jeon, J. Zhang and Z.-H. Zhan, Bi-directional feature fixation-based particle swarm optimization for large-scale feature selection, *IEEE Trans. Big Data* **9**(3) (2022) 1004–1017.

[16] M. Abdel-Salam, G. Hu, E. Çelik, F. S. Gharehchopogh and I. M. El-Hasnony, Chaotic rime optimization algorithm with adaptive mutualism for feature selection problems, *Comput. Biol. Med.* **179** (2024) 108803.

[17] X. Zhou, W. Gui, A. A. Heidari, Z. Cai, G. Liang and H. Chen, Random following ant colony optimization: Continuous and binary variants for global optimization and feature selection, *Appl. Soft Comput.* **144** (2023) 110513.

[18] S. Deng, Y. Li, J. Wang, R. Cao and M. Li, A feature-thresholds guided genetic algorithm based on a multi-objective feature scoring method for high-dimensional feature selection, *Appl. Soft Comput.* **148** (2023) 110765.

[19] P. Wang, B. Xue, J. Liang and M. Zhang, Feature selection using diversity-based multi-objective binary differential evolution, *Inf. Sci.* **626** (2023) 586–606.

[20] X. Wang, F. Wang, Q. He and Y. Guo, A multi-swarm optimizer with a reinforcement learning mechanism for large-scale optimization, *Swarm Evol. Comput.* **86** (2024) 101486.

[21] J. Zhu, J. Liu, Y. Chen, X. Xue and S. Sun, Binary restructuring particle swarm optimization and its application, *Biomimetics* **8**(2) (2023) 266.

[22] X.-f. Song, Y. Zhang, D.-w. Gong and X.-y. Sun, Feature selection using bare-bones particle swarm optimization with mutual information, *Pattern Recognit.* **112** (2021) 107804.

[23] B. Chen, H. Chen and M. Li, Feature selection based on bp neural network and adaptive particle swarm algorithm, *Mob. Inf. Syst.* **2021**(1) (2021) 6715564.

[24] S. Tijjani, M. N. Ab Wahab and M. H. M. Noor, An enhanced particle swarm optimization with position update for optimal feature selection, *Expert Syst. Appl.* **247** (2024) 123337.

[25] M. A. Awadallah, M. S. Braik, M. A. Al-Betar and I. Abu Doush, An enhanced binary artificial rabbits optimization for feature selection in medical diagnosis, *Neural Comput. Appl.* **35**(27) (2023) 20013–20068.

[26] M. S. Braik, M. A. Awadallah, O. Dorgham, H. Al-Hiary and M. A. Al-Betar, Applications of dynamic feature selection based on augmented white shark optimizer for medical diagnosis, *Expert Syst. Appl.* **257** (2024) 124973.

[27] M. A. Awadallah, A. I. Hammouri, M. A. Al-Betar, M. S. Braik and M. Abd Elaziz, Binary horse herd optimization algorithm with crossover operators for feature selection, *Comput. Biol. Med.* **141** (2022) 105152.

[28] M. A. Awadallah, M. A. Al-Betar, M. S. Braik, A. I. Hammouri, I. A. Doush and R. A. Zitar, An enhanced binary rat swarm optimizer based on local-best concepts of pso and collaborative crossover operators for feature selection, *Comput. Biol. Med.* **147** (2022) 105675.

[29] K. Tang, X. Yáo, P. N. Suganthan, C. MacNish, Y.-P. Chen, C.-M. Chen and Z. Yang, Benchmark functions for the CEC'2010 special session and competition on large-scale global optimization, Technical Report (Nature Inspired Computation and Applications Laboratory, USTC, China, 2007), pp. 1–18.

[30] X. Li, K. Tang, M. N. Omidvar, Z. Yang, K. Qin and H. China, Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization, *Gene* **7**(33) (2013) 8.

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

[32] R. Lan, Y. Zhu, H. Lu, Z. Liu and X. Luo, A two-phase learning-based swarm optimizer for large-scale optimization, *IEEE Trans. Cybern.* **51**(12) (2020) 6284–6293.

[33] D. Li, W. Guo, A. Lerch, Y. Li, L. Wang and Q. Wu, An adaptive particle swarm optimizer with decoupled exploration and exploitation for large scale optimization, *Swarm Evol. Comput.* **60** (2021) 100789.

[34] M. R. Bonyadi and Z. Michalewicz, Stability analysis of the particle swarm optimization without stagnation assumption, *IEEE Trans. Evol. Comput.* **20**(5) (2015) 814–819.

[35] Q. Yang, G.-W. Song, W.-N. Chen, Y.-H. Jia, X.-D. Gao, Z.-Y. Lu, S.-W. Jeon and J. Zhang, Random contrastive interaction for particle swarm optimization in high-dimensional environment, *IEEE Trans. Evol. Comput.* **28** (2023) 933–949.

[36] Q. Yang, W.-N. Chen, J. Da Deng, Y. Li, T. Gu and J. Zhang, A level-based learning swarm optimizer for large-scale optimization, *IEEE Trans. Evol. Comput.* **22**(4) (2017) 578–594.

[37] M. N. Omidvar, M. Yang, Y. Mei, X. Li and X. Yao, Dg2: A faster and more accurate differential grouping for large-scale black-box optimization, *IEEE Trans. Evol. Comput.* **21**(6) (2017) 929–942.

[38] X. Ma, Z. Huang, X. Li, L. Wang, Y. Qi and Z. Zhu, Merged differential grouping for large-scale global optimization, *IEEE Trans. Evol. Comput.* **26**(6) (2022) 1439–1451.

[39] A. A. Hadi, A. W. Mohamed and K. M. Jambi, LSHADE-SPA memetic framework for solving large-scale optimization problems, *Complex Intell. Syst.* **5** (2019) 25–40.

[40] I. Guyon, Madelon, UCI Machine Learning Repository (2008), https://doi.org/10.24432/C5602H.
[41] C. Sakar, G. Serbes, A. Gunduz, H. Nizam and B. Sakar, Parkinson's Disease Classification, UCI Machine Learning Repository (2018), https://doi.org/10.24432/C5MS4X.
[42] Ş. Gül and F. Rahim, Period Changer, UCI Machine Learning Repository (2022), https://doi.org/10.24432/C5B31D.
[43] Ş. Gül and F. Rahim, Toxicity, UCI Machine Learning Repository (2022), https://doi.org/10.24432/C59313.
[44] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang and H. Liu, Feature selection: A data perspective, *ACM Comput. Surv.* **50**(6) (2018) 94.
[45] D. W. Aha, D. Kibler and M. K. Albert, Instance-based learning algorithms, *Mach. Learn.* **6** (1991) 37–66.

Minchong Chen received the B.E. degree in automation from Guangxi University of Science and Technology, Liuzhou, China, in 2017. He was a visiting student at Hochschule Magdeburg-Stendal, German, during his third year of undergraduate studies.

He is currently pursuing the M.S. degree at the College of Electronic and Information Engineering, Tongji University, Shanghai, China. His research interests include machine learning, evolutionary computation and its applications.

Hong Li received the B.E. degree in electrical engineering from Jinan University, Guangzhou, China, in 2020. He is currently pursuing the M.S. degree at Tongji University, Shanghai, China.

His current research interests are network topology optimization and computer vision.

Jiwei Tu received the B.E. degree in automation from the Wuhan Institute of Technology, Wuhan, China, in 2020. He is currently pursuing the M.S. degree at Tongji University, Shanghai, China.

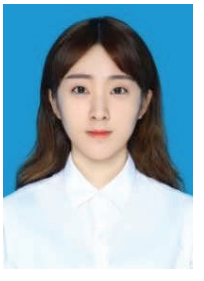
His current research interests include evolutionary computation, constrained multiobjective optimization, machine learning and its application.

Bin He (Member, IEEE) received the B.S. degree in engineering machinery from Jilin University, Changchun, China, in 1996, and the Ph.D. degree in mechanical and electronic control engineering from Zhejiang University, Hangzhou, China, in 2001. From 2001 to 2003, he held post-doctoral research appointments with the State Key Laboratory of Fluid Power Transmission and Control, Zhejiang University.

He is currently a Professor with the Department of Control Science and Engineering, Tongji University, Shanghai, China, and the Shanghai Research Institute for Intelligent Autonomous Systems, Shanghai. His current research interests include intelligent robot control, neural networks, biomimetic microrobots, image processing and fusion, wireless communications, and wireless networks.

Zhen Yin (Member, IEEE) received the B.S. degree in mechanics and engineering science from Fudan University, Shanghai, China, the M.S. degree in mechanical engineering from Rochester Institute of Technology, Rochester, USA, in 2014, and the Ph.D. degree in mechanical engineering from McGill University, Montreal, Canada, in 2020.

He is currently a Professor with the Department of Control Science and Engineering, College of Electronics and Information Engineering, Tongji University. His current research interests include physical intelligence and bioinspired robotics.



Xuejing Hou Received the M.S. degree in Transportation Engineering from the College of Transportation, Dalian Maritime University, Dalian, China, in 2019.

She is currently pursuing her Ph.D. degree at Tongji University, Shanghai, China. She is interested in optimization algorithms design, aviation operations modelling and optimization.

Qi Yu Received the B.E. degree in Traffic Engineering from the College of Transportation, Shanghai Maritime University, Shanghai, China, in 2017.

He is currently pursuing the M.S. degree in Intelligent Transportation Systems Engineering at the College of Transportation, Tongji University, Shanghai, China. His research interests include deep learning, reinforcement learning, and heuristic algorithms.