Information Sciences 633 (2023) 321–342

Contents lists available at ScienceDirect

Information Sciences

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

Heterogeneous cognitive learning particle swarm optimization for large-scale optimization problems

En Zhang [a,b,1], Zihao Nie [a,b,1], Qiang Yang [c,*], Yiqiao Wang [d], Dong Liu [a,b], Sang-Woon Jeon [e], Jun Zhang [e]

[a] College of Computer and Information Engineering, Henan Normal University, Xinxiang, China
[b] Key Laboratory of Artificial Intelligence and Personalized Learning in Education of Henan Province, Xinxiang, China
[c] School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing, China
[d] School of Management, Beijing University of Chinese Medicine, Beijing, China
[e] Department of Electrical and Electronic Engineering, Hanyang University, Ansan, Republic of Korea

A R T I C L E I N F O

Keywords:
Heterogeneous cognitive learning
Particle swarm optimization
Random elite cognitive learning
Stochastic dominant cognitive learning
Large-scale optimization

A B S T R A C T

Large-scale optimization problems (LSOPs) become increasingly ubiquitous but complicated in real-world scenarios. Confronted with such sophisticated optimization problems, most existing optimizers dramatically lose their effectiveness. To tackle this type of problems effectively, we propose a heterogeneous cognitive learning particle swarm optimizer (HCLPSO). Unlike most existing particle swarm optimizers (PSOs), HCLPSO partitions particles in the current swarm into two categories, namely superior particles (*SP*) and inferior particles (*IP*), based on their fitness, and then treats the two categories of particles differently. For inferior particles, this paper devises a random elite cognitive learning (RECL) strategy to update each one with a random superior particle chosen from *SP*. For superior particles, this paper designs a stochastic dominant cognitive learning (SDCL) strategy to evolve each one by randomly selecting one guiding exemplar from *SP* and then updating it only when the selected exemplar is better. With the collaboration between these two learning mechanisms, HCLPSO expectedly evolves particles effectively to explore the search space and exploit the found optimal zones appropriately to find optimal solutions to LSOPs. Furthermore, to help HCLPSO traverse the vast search space with promising compromise between intensification and diversification, this paper devises a dynamic swarm partition scheme to dynamically separate particles into the two categories. With this dynamic strategy, HCLPSO gradually switches from exploring the search space to exploiting the found optimal zones intensively. Experiments are executed on the publicly acknowledged CEC2010 and CEC2013 LSOP benchmark suites to compare HCLPSO with several state-of-the-art approaches. Experimental results reveal that HCLPSO is effective to tackle LSOPs, and attains considerably competitive or even far better optimization performance than the compared state-of-the-art large-scale methods. Furthermore, the effectiveness of each component in HCLPSO and the good scalability of HCLPSO are also experimentally verified.

Abbreviations: HCLPSO, Heterogeneous cognitive learning particle swarm optimization; PSO, Particle swarm optimization; LSGO, Large-scale global optimization; RECL, Random elite cognitive learning; SDCL, Stochastic dominant cognitive learning.

* Corresponding author.
E-mail address: mmmyq@126.com (Q. Yang).

[1] En Zhang and Zihao Nie contributed equally to this paper and thus they are co-first authors.

https://doi.org/10.1016/j.ins.2023.03.086
Received 30 July 2022; Received in revised form 4 March 2023; Accepted 7 March 2023
Available online 11 March 2023
0020-0255/

1. Introduction

Particle swarm optimization (PSO) [1,2] has been experimentally demonstrated to cope with various optimization problems effectively and advanced a lot since it was first devised by Kennedy and Eberhart in 1995 [3]. By virtue of its independence on mathematic properties of optimization problems (like continuous, derivable, and differentiable), easiness in implementation, and strong global search ability [1,4,5], PSO has been frequently utilized to tackle practical optimization problems in real-world engineering.

Specifically, PSO maintains a swarm of particles to traverse the search space to find optimal solutions to optimization problems. Each particle in the swarm is expressed by two vectors, that is, the position vector and the velocity vector. With the two vectors, particles are evolved iteratively in the following manner to fly through the solution space [4]:

$$v_i^d(t+1) \leftarrow wv_i^d(t) + c_1r_1(pbest_i^d(t) - x_i^d(t)) + c_2r_2(gbest^d(t) - x_i^d(t)) \tag{1}$$

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

where $\boldsymbol{x_i}(t) = [x_i^1, \cdots, x_i^d, \cdots, x_i^D]$ and $\boldsymbol{v_i}(t) = [v_i^1, \cdots, v_i^d, \cdots, v_i^D]$ represent the position and the velocity of the ith particle in the tth generation, respectively. $\boldsymbol{pbest_i}(t) = [pbest_i^1, \cdots, pbest_i^d, \cdots, pbest_i^D]$ and $\boldsymbol{gbest}(t) = [gbest^1, \cdots, gbest^d, \cdots, gbest^D]$ represent the personal best position of the ith particle and the global best position of the whole swarm in the tth generation, respectively. w denotes the inertia weight. r_1 and r_2 are two real values randomly sampled within [0,1], while c_1 and c_2 are two acceleration scalars.

In Eq. (1), the second component of the right hand is usually referred to as the cognitive learning part, where a particle generally learns from its own historical experience [4,6]. The third part of the right hand is generally called the social learning part, where a particle usually learns from the historical experience of the whole swarm [4,6]. In the classical PSO, all particles share the same guiding exemplar (namely ***gbest***) in the social learning part. Such an exemplar is too greedy, and therefore, those PSO variants with this learning strategy shown in Eq. (1) are usually confronted with premature convergence or falling into local regions especially when handling multimodal problems [2,6].

To promote the optimization ability of PSO in dealing with multimodal problems, a lot of researchers have devised an ocean of effective learning schemes [6–8] to improve the learning abilities of particles, like the differential elite learning [7], the predominant cognitive learning [8], .etc. Even though these novel learning schemes have assisted PSO to obtain very excellent performance in tackling optimization problems, they are especially designed for small-scale (or low-dimensional) optimization problems. In face of large-scale optimization problems (LSOPs), their optimization effectiveness degrades drastically on account of the "curse of dimensionality" [9].

In particular, different from small-scale optimization problems, the solution space of LSOPs is enlarged exponentially, leading to that searching for optimal solutions in such vast space is considerably difficult like finding a needle in an ocean [10,11]. In addition, in the large-scale space, innumerable local zones or saddle areas exist. As a consequence, premature convergence or falling into local areas quite frequently occurs when PSO is used to tackle LSOPs [12,13]. These difficulties heavily challenge the efficiency and effectiveness of most existing PSO approaches. However, LSOPs are becoming increasingly ubiquitous thanks to the advancement of the Internet of Things and big data [14,15]. As a consequence, there exists an increasing demand for efficient and effective optimizers for large-scale optimization.

To solve LSOPs with high effectiveness and efficiency, researchers have poured extensive attention to designing new effective updating strategies for PSO to cope with LSOPs. Hence, a lot of effective and remarkable large-scale PSO variants have emerged [10,16,17]. Roughly speaking, existing research on large-scale PSOs mainly proceeds in two directions, namely devising holistic large-scale PSO variants [9,18–20], and designing cooperative co-evolutionary large-scale PSO approaches [21,22].

Like traditional PSOs, holistic large-scale PSO variants still treat all variables together and optimize them as a whole [11,23,24]. Since the learning strategy takes the most crucial part in aiding PSO to obtain satisfactory performance, the research on holistic large-scale PSOs mainly focuses on developing effective updating strategies with high diversity for particles, so that the swarm could traverse the solution space thoroughly to locate the optimal solutions. Along this line, many prominent learning schemes have been proposed, such as competitive learning [18], social learning [25], level-based learning [9], two-phase learning [11], stochastic dominant learning [24], etc. Among these distinguished learning strategies, it is interestingly found that most of them discard the use of historical best positions (like ***pbest*** and ***gbest***), but turn to using predominant members in the current swarm to direct the evolution of particles. This is mainly because historical positions are not conducive to promoting the swarm diversity since they might keep unchanged for many iterations during the evolution [11,18,25]. However, particles are generally updated iteration by iteration, and hence, directly using predominant particles to evolve the swarm is of great help to elevate swarm diversity, which is not only profitable for particles to traverse dispersedly to fully explore the search space, but also brings enormous benefit for particles to escape from local zones [9,11,24].

Different from holistic large-scale PSOs, cooperative co-evolutionary PSO (CCPSO) methods primarily utilize the thought of "divide and conquer" to decompose an LSOP into multiple exclusive small-scale sub-problems. Subsequently, these sub-problems are optimized separately by traditional PSOs devised for small-scale optimization problems [26,27]. In general, the optimization process of correlated variables intertwines with each other. As a consequence, the most vital part of CCPSO is to decompose an LSOP into exclusive low-dimensional sub-problems accurately, so that interacting variables are put into the same low-dimensional problem to optimize [22,28]. To achieve this end, the research on CCPSO principally concentrates on designing effective decomposition mechanisms by detecting variable interdependencies. As a result, multifarious and outstanding decomposition methods have been proposed, such as recursive differential grouping (RDG) [29], merged differential grouping (MDG) [30], efficient adaptive differential grouping (EADG) [31], etc.

The above-mentioned PSO methods have been experimentally verified to attain very promising optimization performance in coping with certain kinds of LSOPs. Nevertheless, they are still up against being trapped into locally optimal zones and premature convergence in coping with complex LSOPs, especially the ones with highly complicated landscapes resulting from too many interacting variables. Therefore, how to elevate the optimization effectiveness of PSO in handling complicated LSOPs still needs further research. To achieve this goal, we propose a heterogeneous cognitive learning PSO (HCLPSO) to tackle LSOPs.

Specifically, different from most existing PSO variants which treat all particles equally, HCLPSO treats different particles differently by first separating them into two categories, namely superior particles (***SP***) and inferior particles (***IP***), according to their fitness. Then, two different learning strategies are devised to update the two categories of particles, respectively. In this way, we aim to let inferior particles concentrate on exploring the solution space to find optimal zones, and let superior particles focus on exploiting the optimal areas where they locate to find high-quality solutions. As a result, HCLPSO is expected to maintain a good balance between exploration and exploitation to obtain satisfactory optimization performance in solving LSOPs. It should be mentioned that HCLPSO belongs to the first category of large-scale PSO variants, namely the holistic large-scale PSOs, since it considers all variables as a whole to optimize LSOPs.

To make a summary, the main contributions of this paper are summed up as follows:

1) A heterogeneous cognitive learning strategy is devised to update the two categories of particles differently. Concretely, a random elite cognitive learning (RECL) scheme is devised to evolve each inferior particle by randomly selecting a superior particle from ***SP*** as the guidance exemplar. By contrast, a stochastic dominant cognitive learning (SDCL) strategy is designed to evolve each superior particle by first randomly selecting an exemplar from ***SP*** and then updating it only when the chosen exemplar is better. By this means, inferior particles learn from diverse but better exemplars to move toward promising regions, while those updated superior particles learn from predominant peer companions to find high-quality solutions. Therefore, the cooperation between these two learning schemes endows a good compromise between fast convergence and search diversity for HCLPSO to explore the search space and exploit the found promising regions appropriately.
2) An adaptive swarm partition strategy is further designed to divide particles in the current swarm into the two categories dynamically. To be specific, a non-linear decreasing function regarding the number of used fitness evaluations is designed to dynamically adjust the number of superior particles during the evolution. With this dynamic strategy, superior particles become fewer and fewer, while inferior particles become more and more as the evolution proceeds. As a result, HCLPSO gradually switches from exploring the search space diversely to exploiting the found optimal regions intensively.

By means of the collaboration between the above two techniques, HCLPSO expectedly evolves particles effectively and efficiently to traverse the large-scale solution space. Therefore, HCLPSO is hopefully promising for solving LSOPs. To substantiate its optimization effectiveness and efficiency, this paper carries out a lot of comparative experiments on the publicly acknowledged CEC2010 [32] and CEC2013 [33] LSOP suites by comparing it with several state-of-the-art large-scale approaches. Additionally, investigation experiments are also executed to testify the effectiveness of each component in HCLPSO, and its good scalability to solve LSOPs with different dimensionalities.

The rest structure of this article is arranged as follows. Section 2 introduces closely related studies on large-scale PSOs. Section 3 elucidates HCLPSO in detail. Section 4 affords comprehensive experimental verification of the efficiency and effectiveness of HCLPSO on two commonly adopted benchmark LSOP sets. At last, the conclusion of this paper is given in Section 5.

2. Related work on large-scale PSO

Due to the “curse of dimensionality”, most existing PSO variants attain unsatisfactory optimization performance when dealing with LSOPs. To improve the optimization capability of PSO in coping with such problems, many researchers have developed a lot of outstanding large-scale PSO variants from the following two aspects.

2.1. Holistic large-scale PSOs

The same with traditional PSOs, holistic large-scale PSO variants still evolve all variables together to find the optimal solutions to LSOPs [18,19,25]. On account of the aforementioned challenges of LSOPs, the key to finding as accurate solutions to LSOPs as possible for this kind of large-scale PSOs is to locate optimal regions efficiently in the vast solution space. To this end, high diversity maintenance is usually required [17]. Along this road, a large number of remarkable and effective learning mechanisms have been proposed for PSO to boost the learning abilities of particles. In the following, we review some representative and typical ones.

In [18], Cheng *et al.* designed a competitive swarm optimizer (CSO) by introducing a pairwise competition technique into particles. To be specific, this optimizer randomly organizes particles into pairs and subsequently compares the two members in each pair. Then, it lets the loser learn from the associated winner and the center of the entire swarm, while it lets the winner directly enter the next generation. In [25], Cheng *et al.* devised a social learning PSO (SLPSO) by first ranking particles from the smallest to the largest with respect to their fitness and then probabilistically updating each inferior particle by learning from a random predominant one in the swarm and the center of the whole swarm. In [34], a region encoding technique was devised along with an adaptive region search strategy. Then, the two techniques were embedded into SLPSO to further elevate its optimization effectiveness in tackling LSOPs. In [19], a segment-based predominant learning PSO (SPLSO) was developed by first randomly partitioning the position vector and the velocity vector of each updated particle into multiple equally sized segments. Subsequently, it employs different relatively better particles to update different segments of this particle. By this means, each updated particle learns from multiple predominant ones in the swarm. Considering the fact that particles in the swarm likely preserve distinct strengths in exploring the search space and exploiting the found optimal regions, the authors of [9] devised a level-based learning PSO (LLSO) to treat particles differently. To be specific, this PSO variant divides particles in the current swarm into different layers on the basis of their fitness values. Next, each particle in lower layers is evolved by learning from those randomly chosen from higher layers. To adaptively adjust the number of levels in LLSO, Wang *et al*. [23] designed a reinforcement learning algorithm to dynamically adjust the number of levels in LLSO, leading to a reinforcement learning level PSO (RLLPSO). In [35], a ranking-based bias learning PSO (RBLSO) was designed by updating poor particles with predominant ones and the weighted center of the swarm based on a ranking paired learning scheme and a biased center learning strategy. In [36], a multi-level sampling scheme was devised to first divide particles into multi-levels on the basis of their fitness, and then randomly select particles from each level to update. Then, a dynamic p-learning scheme was developed to update each selected particle by using a randomly selected exemplar from the top p (which is dynamically adjusted for each particle during the evolution) best ones of all selected particles and the weighted center of all these particles. In [37], Li *et al.* first devised an explorative learning mechanism and an adaptive exploitative learning scheme to dynamically select two separate groups of particles to be updated and their associated guiding exemplars. Then, only particles in the intersection set of the two groups are updated by learning from their associated guiding exemplars selected by the two learning mechanisms.

Instead of utilizing only one unified learning scheme to update particles, some researchers have even proposed to employ multiple learning schemes to evolve particles, so as to elevate the learning effectiveness of particles. For instance, in [11], a two-phase learning PSO (TPLSO) was devised by designing different learning strategies for particles in different evolution processes. Concretely, this PSO method separates the evolution process into the elite learning process and the mass learning process. In the mass learning process, particles are arranged into triads and then a competitive learning scheme is adopted to evolve particles in each triad. In the elite learning process, the top ranked elites are chosen and then they are updated by learning from each other to exploit the optimal regions they locate at. [38] proposed a multi-policy learning PSO (MSL-PSO) by also utilizing different learning mechanisms to evolve particles in different evolutionary process. Concretely, in the first process, particles are evolved by learning from their associated dominators and the center of the swarm. Then, in the second process, particles are further guided by the historically probed best positions and the best positions in the current swarm.

The above large-scale PSO variants have experimentally demonstrated to be effective in solving certain kinds of LSOPs. Nevertheless, experimental results in the above studies [35–37] have also shown that most of them have limitations in solving complicated LSOPs with complex properties, like overlapping interacting variables, and fully non-separable variables as shown in the CEC2010 [32] and the CEC2013 [33] benchmark LSOP sets. Therefore, the research on this category of large-scale PSO methods still deserves further dedication.

2.2. Cooperative co-evolutionary large-scale PSOs

Cooperative co-evolutionary PSOs (CCPSOs) [27] mainly combine the cooperative co-evolution framework (CC) [39] with PSO to solve LSOPs. The key thought of CC is to utilize the "divide and conquer" [26] strategy to first divide an LSOP into several small-scale sub-problems and then optimize each sub-problem separately to find the optimal solution. In [21], Bergh and Engelbrecht took the first try to combine CC with PSO to design CPSO-S_K and CPSO-H_K by randomly dividing the dimensions of an LSOP into K sub-problems. In particular, CPSO-S_K optimizes each sub-problem separately, while CPSO-H_K alternatively adopts CPSO-S_K and the classical PSO to optimize LSOPs. Subsequently, instead of using fixed K in CPSO-S_K, Li and Yao developed an improved version of CPSO-S_K, named CCPSO2 [22]. Particularly, this optimizer maintains a pool of the number of sub-problems and then dynamically changes the number of the decomposed sub-problems by randomly selecting a number from the pool.

Since the divided sub-problems in CCPSOs are individually optimized, the effectiveness of CCPSO heavily depends on the decomposition accuracy of the sub-problems for an LSOP because the optimization of correlated variables generally intertwines with each other [26]. Therefore, recent research on CC-based large-scale optimizers mainly concentrates on designing effective decomposition schemes.

Along this line, the most representative decomposition method is the differential grouping (DG) algorithm [40], which first detects the correlation between pairwise variables based on the partial difference with respect to fitness values after shifting the settings of the two variables. However, DG cannot identify indirectly related variables. To fill this shortcoming, a global version of DG (GDG) [41] was proposed. This method keeps a correlation matrix and fills this matrix with zeros and ones according to the detected pairwise variable correlations based on DG with zero denoting that the two variables are independent while one denoting that they are interacting. Then, based on the correlation matrix, the variables are separated into sub-problems by placing all interacting variables together.

Nevertheless, the above DG variants consume too many fitness evaluations ($O(D^2)$ with D denoting the dimension size) to detect the dependencies among variables. This leads to that the optimization of sub-problems is not enough to get high-quality solutions to LSOPs. To conquer this predicament, an improved variant of DG, called DG2 [42], was designed, which saves a lot of computational resources by repeatedly using sampled points to detect the relationship of any two variables. Ma *et al.* [30] designed a merged DG (MDG) by employing the thought of binary search to detect the interaction between variable subsets. Particularly, this decomposition method first identifies variables into two sets, namely the separable variable set and the non-separable variable set. Next, the non-separable variables are further separated into several sub-sets by a binary-tree-based iterative merging algorithm. By this means, a lot of fitness evaluations are saved. Inspired by the binary search as well, a recursive DG (RDG) [29] was devised where detecting variable dependencies only takes O(Dlog(D)) objective function evaluations. To further promote the detection efficiency of RDG, Yang *et al.* [43] developed an efficient RDG (ERDG) by reusing the historical information on detecting correlations between variables, which saves a lot of fitness evaluations by avoiding detecting some interactions between variables. In [44], a hybrid deep grouping approach was designed where both the interdependence between variables and the essentialness of variables are considered to decompose them into sub-problems. In [45], Zhang *et al.* devised a dynamic decomposition method according to the contribution of variables, which is measured by the historical evolutionary information. In each generation of the evolution, variables are dynamically decomposed into sub-problems based on the measured contributions and correlation information.

Except for the above two categories of large-scale PSOs, a few researchers have even designed distributed large-scale PSO variants. For instance, a dynamic group learning distributed PSO (DGLDPSO) was designed in [46], where the entire population is partitioned into several sub-populations and then the master–slave distribution model is adopted to evolve these sub-populations simultaneously. In [27], an adaptive granular learning distributed PSO (AGLDPSO) was developed, which also adopts the master–slave distribution model to coevolve multiple sub-populations. Particularly, in this method, a locality-sensitive hashing based clustering method was devised to partition the whole population into sub-population, and a logistic regression based adaptive granular learning scheme was developed for dynamically determining the sub-population size. In [47], a distributed elite guided learning swarm optimizer (DEGLSO) was devised to maintain multiple swarms and then adopt the master–slave distributed model to cooperatively evolve these swarms in parallel to find the optimal solutions to LSOPs.

Although existing large-scale PSO variants have exhibited good optimization performance in dealing with certain types of LSOPs, they are still up against many limitations in coping with complex LSOPs, particularly the ones with many non-separable variables and numerous wide local regions. As a result, the global optima found by most existing large-scale PSOs are far from the true global optima of complicated LSOPs. This indicates that the optimization capability of PSO in tackling LSOPs still needs promotion. To this end, we propose a heterogeneous cognitive learning PSO (HCLPSO) in this paper by devising different learning schemes for different kinds of particles so as to elevate their searching effectiveness in the huge solution space.

3. Heterogeneous cognitive learning particle swarm optimization

During the continuous evolution of particles, different particles likely own different advantages in exploring the vast search space for locating promising regions and exploiting the found optimal zones for finding high-quality solutions. In particular, compared with inferior particles, superior ones preserve higher probabilities in locating at or around the optimal regions in the solution space. Therefore, they should focus on exploiting the optimal regions where they locate to get as accurate solutions as possible. In contrast, it is likely that inferior particles are scattered more dispersedly in different zones than superior ones. Hence, inferior particles should concentrate on exploring the huge solution space to locate more optimal regions.

Bearing the above inspiration into mind, we propose a heterogeneous cognitive learning PSO (HCLPSO) by designing different learning policies for different kinds of particles so as to let the two kinds of particles take different responsibilities to explore the immense search range and exploit the found optimal regions. To be specific, we first partition particles in the swarm into two categories, namely the superior particles (***SP***) and the inferior particles (***IP***), based on their fitness. That is, given the swarm size is NP and the number of superior particles is NSP, ***SP*** contains the top best NSP particles, while ***IP*** consists of the rest (NP-NSP) particles. Subsequently, we design two different learning strategies to heterogeneously update the two kinds of particles, which are elucidated in the following.

3.1. Heterogeneous cognitive learning

3.1.1. Random elite cognitive learning for inferior particles

In general, in comparison with inferior particles, on the one side, superior particles are expectedly closer to optimal regions; on the other side, they contain more valuable evolutionary information to evolve the swarm. Therefore, they can be employed to lead the update of inferior particles to move toward promising areas. To achieve this goal, we propose a random elite cognitive learning (RECL) strategy for inferior particles by using superior particles as their guiding exemplars.

On the one hand, inferior particles are not only usually scattered more dispersedly in different zones in the solution space, but also generally farther from the globally optimal areas than superior particles. Therefore, they prefer to select better exemplars to learn from so that they can approach optimal areas fast. On the other hand, among the superior particles, the better fitness value one particle owns, the higher probability it may have that it is close to optimal areas. Therefore, to guide inferior particles to move to optimal regions fast, superior particles with better fitness should have higher selection probabilities.

Bearing the above considerations into mind, instead of using uniform selection to choose superior particles to direct the evolution of inferior ones, we design a non-linear weight function in the following to calculate the selection probabilities of superior particles:

$$w_j = \frac{1}{\sigma NSP\sqrt{2\pi}} e^{-\frac{(rank(j)-1)^2}{2\sigma^2 NSP^2}} \tag{3}$$

where w_j is the weight of the jth superior particle in ***SP***, $rank(j)$ is its rank after all superior particles are ranked from the smallest to the largest with respect to their fitness, NSP is the number of superior particles, and σ is a control parameter that has a significant impact on the weight. In this paper, σ is set to 0.1. The investigation of this parameter setting is conducted in the experiments in Section 4.3.

After obtaining the weights of superior particles, the selection probabilities of these particles are computed as follows:

$$p_j = \frac{w_j}{\sum_{i=1}^{NSP} w_i} \tag{4}$$

where p_j is the selection probability of the jth superior particle.

With the above calculated selection probabilities, RECL randomly selects a superior particle from $\boldsymbol{SP}$ based on the roulette wheel selection method for each inferior particle. Then, the inferior particle is updated as follows:

$$\boldsymbol{v}_{IP,i}(t+1) \leftarrow r_1 \times \boldsymbol{v}_{IP,i}(t) + r_2 \times \beta \times (\boldsymbol{x}_{SP,rand1}(t) - \boldsymbol{x}_{IP,i}(t)) \tag{5}$$

$$\boldsymbol{x}_{IP,i}(t+1) \leftarrow \boldsymbol{x}_{IP,i}(t) + \boldsymbol{v}_{IP,i}(t+1) \tag{6}$$

where $\boldsymbol{v}_{IP,i}$ and $\boldsymbol{x}_{IP,i}$ denote the velocity and the position of the ith inferior particle in $\boldsymbol{IP}$, respectively. $\boldsymbol{x}_{SP,rand1}$ represents the randomly selected superior particle from $\boldsymbol{SP}$ by the roulette wheel selection method with $rand1 \in [1, NSP]$ denoting its index. r_1 and r_2 denote two real random vectors with each element uniformly generated from [0,1]. β represents a controlling factor used to take charge of the learning degree from the selected exemplar.

From Eqs. (3) ~ (6), the following observations can be attained:

1) Each inferior particle learns from a random superior particle, which is randomly chosen from $\boldsymbol{SP}$. On the one hand, each inferior particle is ensured to learn from a better one, which is likely close to optimal areas. By this means, inferior particles expectedly move toward optimal regions. This is conducive to locating optimal solutions fast. On the other hand, since the guiding exemplar of each inferior particle is randomly selected, different inferior particles may have different guiding exemplars. This is conducive to generating diversified offspring for inferior particles to seek for promising regions in different directions. By virtue of this mechanism, high exploration diversity is expectedly maintained for inferior particles to traverse the search space to locate more promising regions.
2) Regarding the selection of superior particles, it is found that superior particles with better fitness values have exponentially higher selection probabilities to be chosen as the guiding exemplars of the inferior ones. Therefore, it is likely that inferior particles are directed by those superior ones with much better fitness in $\boldsymbol{SP}$. In this way, the approaching speed of inferior particles to optimal regions can be accelerated. Nevertheless, it deserves notice that such acceleration is achieved without rapid sacrifice of search diversity, thanks to that the directing exemplars of inferior particles are randomly chosen from $\boldsymbol{SP}$.

In short, with the designed RECL strategy, the learning efficiency and effectiveness of inferior particles can be guaranteed with a potential balance between intensification and diversification. As a result, inferior particles can explore the huge solution space with slight intensification to seek for as many promising regions as possible. Experiments carried out in Section 4.3 will verify the effectiveness of this learning strategy.

3.1.2. Stochastic dominant cognitive learning for superior particles

Different from inferior particles, superior particles are generally close to or even locate at optimal regions in the solution space. Therefore, they should concentrate on exploiting the regions where they locate to refine the solution accuracy. At the same time, however, local convergence should be avoided. To reach this end, we design a stochastic dominant cognitive learning (SDCL) strategy for superior particles.

Specifically, SDCL works as follows. For each superior particle ($\boldsymbol{x}_{SP,i}$), at first, another superior particle (suppose it is $\boldsymbol{x}_{SP,rand2}$) is randomly selected from $\boldsymbol{SP}$. Next, the chosen $\boldsymbol{x}_{SP,rand2}$ is competed with $\boldsymbol{x}_{SP,i}$. Only when the chosen $\boldsymbol{x}_{SP,rand2}$ is no worse than $\boldsymbol{x}_{SP,i}$, $\boldsymbol{x}_{SP,i}$ is evolved by learning from $\boldsymbol{x}_{SP,rand2}$; otherwise, $\boldsymbol{x}_{SP,i}$ remains unchanged and directly enters the next iteration. In particular, when $\boldsymbol{x}_{SP,i}$ is evolved, it is updated as follows:

$$\boldsymbol{v}_{SP,i}(t+1) \leftarrow r_1 \times \boldsymbol{v}_{SP,i}(t) + r_2 \times \beta \times (\boldsymbol{x}_{SP,rand2}(t) - \boldsymbol{x}_{SP,i}(t)) \tag{7}$$

$$\boldsymbol{x}_{SP,i}(t+1) \leftarrow \boldsymbol{x}_{SP,i}(t) + \boldsymbol{v}_{SP,i}(t+1) \tag{8}$$

where $\boldsymbol{v}_{SP,i}$ and $\boldsymbol{x}_{SP,i}$ denote the velocity and the position of the ith superior particle in $\boldsymbol{SP}$, respectively. $\boldsymbol{x}_{SP,rand2}$ represents the randomly selected superior particle from $\boldsymbol{SP}$ based on the uniform distribution with $rand2$ denoting its index. It should be noticed that $\boldsymbol{x}_{SP,rand2}$ is no worse than $\boldsymbol{x}_{SP,i}$ when $\boldsymbol{x}_{SP,i}$ is triggered to update. The meanings of the other parameters (namely r_1, r_2, and β) are the same as those in Eq. (5).

From Eq. (7), the following findings are attained:

1) Each updated superior particle learns from an exemplar with no worse fitness than itself. This ensures that each updated superior particle could move toward another more promising region or exploit the optimal region where it locates to refine its accuracy. In this manner, effective exploitation by the updated superior particles around the optimal regions can be achieved.

2) Due to the stochastic competition, superior particles with relatively poor fitness have chances to survive. As a result, high diversity could be maintained when superior particles intensively exploit optimal regions. This is conducive to the swarm to avoid being fallen into local areas.
3) In particular, it is found that the superior particle with the worst fitness in ***SP*** is always updated, while the one with the best fitness is always not updated. However, it should be noticed that the best superior particle in the current iteration may be replaced by the updated ones in the next iteration. Therefore, with the iteration-by-iteration evolution, superior particles become better and better and at last they expectedly locate at or around the optimal solutions to LSOPs.

As a whole, with the designed SDCL strategy, the learning effectiveness of superior particles can be ensured with fast moving toward more promising regions or intensive exploitation of the areas where they locate to refine solutions. As a result, superior particles can exploit the optimal zones where they locate with relatively high diversity to find as accurate solutions as possible. Experiments carried out in Section 4.3 will verify the effectiveness of this learning mechanism.

Together, with the cohesive cooperation of the above two heterogeneous learning schemes, particles in the swarm are expected to explore the vast search space for locating optimal regions and exploit the found optimal zones for finding high-quality solutions appropriately. As a result, HCLPSO hopefully achieves promising optimization performance in solving LSOPs.

3.2. Adaptive partition for superior and inferior particles

Taking deep investigation on the RECL and SDCL strategies, we find that the partition of the swarm into inferior particles and superior particles, which is controlled by the number of superior particles (*NSP*) in this paper, takes crucial part in these two learning schemes. To be specific, as *NSP* becomes larger and larger, fewer and fewer inferior particles are updated by learning from more and more superior particles. In this situation, both the learning ranges of inferior particles and superior ones are large. This is very beneficial for diversity enhancement, but may lead to slow convergence. In contrast, when *NSP* becomes smaller and smaller, more and more inferior particles are evolved by learning from fewer and fewer superior particles. In this case, the learning ranges of these two kinds of particles are small. This is profitable for the acceleration of the convergence to optimal solutions, but may result in premature convergence.

Based on the above analysis, it is found that in the early evolution, since particles might as well concentrate on exploring the immense solution space to locate optimal regions fast, *NSP* should be large, so that high search diversity can be kept. By contrast, in the late evolution, since particles should concentrate on exploiting the found optimal zones to find high-quality solutions, *NSP* should be small, such that inferior particles learn intensively from fewer superior ones and superior particles exploit the optimal areas they locate at intensively as well.

To achieve the above end, this paper devises the following dynamic adjustment of *NSP*:

$$SPR = SPR_{\max} - (SPR_{\max} - SPR_{\min}) \times (\frac{Fes}{Fes_{\max}})^{0.5} \tag{9}$$

$$NSP = \lfloor SPR \times NP \rfloor \tag{10}$$

where *SPR* is the ratio of superior particles to all particles. It is used here for convenient adjustment of *NSP*, which is the number of superior particles. *Fes* represents the number of fitness evaluations exhausted so far, and $Fes_{\max}$ denotes the preset maximum number of fitness evaluations. $\lfloor x \rfloor$ represents the floor function returning the largest integer but smaller than x. SPR_{max} and SPR_{min} are the maximum and the minimum ratios of superior particles out of all particles.

From the above equations, we find that as the evolution continues, *NSP* gradually decreases from $SPR_{max} \times NP$ to $SPR_{min} \times NP$. Therefore, the learning ranges of both kinds of particles become smaller and smaller. As a result, the swarm gradually switches from exploring the search space to exploiting the found optimal regions. This is conducive to dynamically compromising exploration and exploitation when particles traverse the immense search space to seek for high-quality solutions to LSOPs. In this paper, for the convenience of parameter fine-tuning, we set $SPR_{min} = 0.5 \times SPR_{max}$. Then, SPR_{max} is set as 0.9, and hence SPR_{min} is 0.45. The investigation on the setting of SPR_{max} is conducted in the experiments in Section 4.3.

In conclusion, the adaptive partition strategy provides a special compromise between exploration and exploitation for the swarm from the perspective of the swarm level, and thus expectedly further helps HCLPSO attain good optimization performance in solving LSOPs. The effectiveness of this strategy is verified by experiments carried out in Section 4.3.

3.3. Complete procedure of HCLPSO

Combining the above devised heterogeneous learning strategies and the designed adaptive swarm partition scheme, we get the complete HCLPSO, which is outlined in Algorithm 1.

From Algorithm 1, it is found that HCLPSO mainly contains three steps, namely the swarm partition (Lines 3 ~ 4), the updating of inferior particles (Lines 6 ~ 10), and the updating of superior particles (Lines 11 ~ 17). In the swarm partition stage, all particles are ranked first according to their fitness and then are divided into superior particles and inferior particles according to the calculated *NSP*. Before the updating of inferior particles, the selection probabilities of superior particles are first computed and then each inferior particle is updated by the devised RECL strategy. Then, in the updating of the superior particles, each superior particle is updated by the designed SDCL strategy. The above three main steps iteratively continue until the given fitness evaluations are exhausted.

From Algorithm 1, we can get that the total time complexity of HCLPSO is $O(NP \times D + NP \times logNP)$ in each generation. Specifically, in the swarm partition step, the total time complexity is $O(NP \times logNP)$. In the updating of inferior particles, the total time complexity is $O((NP\text{-}NSP) \times D)$, while in the updating of superior particles, the total time complexity is $O(NSP \times D)$. Together, the total time complexity of updating particles is $O(NP \times D)$ in each iteration. Overall, it is discovered that no serious computational burden is exerted on HCLPSO as compared with most existing PSO variants.

3.4. Difference between HCLPSO and existing large-scale PSO variants

In summary, there are two basic ideas behind HCLPSO. The first one is adopting better particles to direct the update of worse particles. The other is employing heterogeneous learning strategies to update different kinds of particles. With respect to the first idea, HCLPSO shares the same thought with some existing large-scale PSO variants, like CSO [18], SLPSO [25], LLSO [9], and RBLSO [35]. Regarding the second thought, HCLPSO also shares similarity with some existing large-scale PSO methods, such as TPLSO [11] and MSL-PSO [38]. In comparison with these existing PSO variants for LSOPs, HCLPSO distinguishes from them in the following main aspects:

1) The most significant difference between HCLPSO and the above-mentioned large-scale PSO variants lies in that HCLPSO only utilizes one guiding exemplar to direct the update of each particle. Nevertheless, in these existing large-scale PSO methods, two guiding exemplars are usually employed to update each particle. Concretely, in CSO, SLPSO, TPLSO, and RBLSO, a random dominator of each updated particle and the mean position (or the biased mean position) of the whole swarm are utilized to direct its update. In particular, the second exemplar, namely the mean position (or the biased mean position) of the current swarm is shared by all particles, which is not conductive to the promotion of swarm diversity. In LLSO and MSL-PSO, two different predominant particles in the current swarm are employed to direct the evolution of each updated particle. In general, compared with the guidance of two exemplars, the guidance of only one exemplar could bring higher search diversity for the swarm. Therefore, theoretically, HCLPSO could preserve higher search diversity than the existing large-scale PSO variants to explore the vast problem space of LSOPs.
2) HCLPSO adopts absolutely better particles to direct the evolution of each particle in ***IP*** by randomly selecting guiding exemplars from ***SP***, while it employs relatively better ones to guide the update of each particle in ***SP*** by using the pairwise competition mechanism. Besides, the selection probability of each superior particle in ***SP*** for inferior particles is different from each other, and the better one superior particle is, the exponentially higher selection probability it has. However, in CSO and TLPSO, the pairwise competition and the triple competition mechanisms are employed to evolve the swarm, respectively, and thus they both utilize relatively better particles to guide the update of relatively worse ones. In SLPSO, RBLSO, LLSO, and MSL-PSO, particles are all sorted from the best to the worst and then each updated particle is evolved by absolutely better ones. In particular, it is found that in all these existing PSO variants, the selection probabilities of the potential guiding exemplars are the same for each updated particle. Whereas, in HCLPSO, the selection probabilities of exemplars in ***SP*** for particles in ***IP*** are different, while the selection probabilities of exemplars are the same for particles in ***SP***. Such a difference implicitly affords different learning abilities for different kinds of particles in HCLPSO, which may bring benefit for the swarm to explore and exploit the vast search space appropriately.
3) HCLPSO utilizes two very different learning schemes to update the two kinds of particles. However, in CSO, SLPSO, LLSO, and RBLSO, one uniform framework is adopted to update all particles. Though different learning schemes are adopted to update particles in TPLSO and MSL-PSO, they are designed from the perspective of different learning phases. In the same learning phase, particles are updated with the same learning strategy. However, the heterogeneous learning schemes in HCLPSO are designed from the particle-level. In other words, for different kinds of particles, different learning strategies are utilized. In this way, HCLPSO could let inferior particles focus on exploring the vast solution space to locate optimal regions fast, while it could let superior particles concentrate on exploiting the optimal areas where they locate to find high-quality solutions.

4. Experiments

This section performs a series of comparative experiments on the publicly acknowledged CEC2010 [32] and CEC2013 [33] LSOP benchmark suites to testify the optimization efficiency and effectiveness of HCLPSO. The CEC2010 suite consists of twenty 1000-D LSOPs with different properties, including unimodal, multimodal, fully separable, fully non-separable, and partially separable. The CEC2013 suite contains fifteen 1000-D LSOPs. They are extended from the LSOPs in the CEC2010 suite via introducing many new properties, like overlapping, and imbalance contributions of variables. Theoretically, the LSOPs in the CEC2013 suite are far more complicated and much harder to optimize than those in the CEC2010 suite. Tables S1 and S2 in the Supplementary material summarize the main properties of these two benchmark test suites respectively. Detailed information about these two LSOP suites can be found in [32] and [33].

Algorithm 1 HCLPSO

Input: swarm size NP, control parameter β, maximum fitness evaluations Fes_{Max}.

Randomly sample values in the domain of each variable to initialize the swarm and compute its fitness; Set $Fes = NP$;

While $Fes < Fes_{Max}$ **do**

(*continued on next page*)

(*continued*)

Algorithm 1 HCLPSO

Rank particles from the smallest to the largest in terms of their fitness;
Calculate the number (*NSP*) of superior particles based on Eq. (9) and Eq. (10);
Compute the selection probability of each superior particle according to Eq. (3) and Eq. (4);
For $i = NSP + 1 : NP$ **do // Update inferior particles**
  Randomly select a superior particle $\boldsymbol{x}_{SP,rand1}$ from ***SP*** according to the roulette wheel selection method;
  Update particle $\boldsymbol{x}_{IP,i}$ by Eq. (5) and Eq. (6);
  Compute the fitness value of the updated particle $\boldsymbol{x}_{IP,i}$ and *Fes*++;
End For;
For $i = 1 : NSP$ **do // Update superior particles**
  Randomly select a particle $\boldsymbol{x}_{SP,rand2}$ from ***SP***;
  If $f(\boldsymbol{x}_{SP,rand2}) \le f(\boldsymbol{x}_{SP,i})$ **then**
    Update particle $\boldsymbol{x}_{SP,i}$ by Eq. (7) and Eq. (8);
    Compute the fitness value of the updated particle $\boldsymbol{x}_{SP,i}$ and *Fes*++;
  End If
End For
End While
Get the best particle $\boldsymbol{x}$ in the swarm and output it along with its fitness;

Output: The found optimal solution $\boldsymbol{x}$ and its fitness $f(\boldsymbol{x})$;

In this section, this paper first carries out parameter investigation experiments on the CEC2010 benchmark suite to find the optimal settings of the control factor β and the number of particles *NP* in Section 4.1. Next, extensive comparisons between HCLPSO and eight state-of-the-art optimizers for LSOPs are made on the CEC2010 and the CEC2013 benchmark suites in Section 4.2. Subsequently, in Section 4.3, deep investigation on HCLPSO is conducted to verify the usefulness of the devised heterogeneous cognitive learning strategies and the designed adaptive swarm partition method. At last, in Section 4.4, we further investigate the scalability of HCLPSO to solve LSOPs with different dimensionalities by comparing with the eight selected large-scale optimizers on the CEC2010 benchmark suite.

To comprehensively verify the optimization effectiveness and efficiency of HCLPSO, we choose eight latest and representative optimization methods for LSOPs to compare with it on the two commonly utilized LSOP suites. In particular, the eight selected large-scale methods are TPLSO [11], LLSO [9], SPLSO [19], SLPSO [25], CSO [18], DECC-DG2 [42], DECC-RDG [29], and DECC-RDG2 [48], respectively. The former five large-scale methods are all PSO variants and belong to the first category of large-scale PSOs as reviewed in Section 2.1. It should be mentioned that HCLPSO also belongs to this category. The latter three large-scale algorithms are CC based large-scale optimization methods, which belong to the second category as reviewed in Section 2.2. It should be mentioned that the three decomposition methods (namely DG2, RDG and RDG2) are state-of-the-art variable grouping approaches and they are combined with DE [49,50] to solve LSOPs instead of PSO here because it has been demonstrated that DE is more effective than PSO to solve LSOPs when combined with the CC framework in the literature [26,29,42,48].

Unless otherwise specified, the maximum number of fitness evaluations is configured to $3000 \times D$ with D denoting the number of variables in the optimization prblems. To comprehensively assess all algorithms, we separately execute each method 30 times, and then adopt the median, the mean, and the standard deviation values over the 30 separate runs to assess its optimization performance. Furthermore, to illustrate the statistical significance, we perform the Wilcoxon rank-sum test at $\alpha = 0.05$ to compare the optimization result of HCLPSO with that of each compared method on each benchmark LSOP. In addition, we also conduct the Wilcoxon signed rank test at $\alpha = 0.05$ to tell whether there is significant difference between the optimization results of HCLPSO and those of each compared method on one whole benchmark set. Besides, we additionally employ the mean value of each method on each LSOP to perform the Friedman test at $\alpha = 0.05$ to get the averaged rank of each algorithm on one whole benchmark suite.

For better presentation and understanding, we attach the detailed comparison results with big tables to the Supplementary material, and only present the summarized statistical comparison results and some investigation results with small tables in this paper. In all tables in this paper and the Supplementary material, the p-values displayed in bold indicate that the optimization performance of HCLPSO is significantly better than those of the corresponding compared approaches on the associated LSOPs based on the Wilcoxon rank-sum test. "+", "−" and "=" marked above the p-values indicate that the optimization results of HCLPSO are significantly better than, significantly worse than, and equivalent to the corresponding compared algorithms on the associated LSOPs. "*w/t/l*" count the numbers of "+", "=", and "−", respectively. "*Rank*" denotes the average rank of each algorithm obtained from the Friedman test.

At last, it deserves mentioning that all algorithms are executed on the same computer with 8 Intel Core i7-10700 2.90-GHz CPUs, 8-GB memory, and Ubuntu 12.04 LTS 64-bit system.

4.1. Parameter setting investigation of HCLPSO

To explore the optimal settings of the control factor β and the number of particles *NP* in HCLPSO, investigation experiments are executed on the CEC2010 LSOP suite via ranging *NP* from 200 to 700 with step size 100, and varying β from 0.75 to 1.00 with step size 0.05. The experimental results are presented in Table S3 in the Supplementary material. To make it clear to see, we highlight the best optimization results in each part attained by HCLPSO with the same *NP* but with different values of β in bold. In particular, two averaged ranking values obtained from the Friedman test are presented. The first averaged rank denoted as "*Rank*1" is obtained by the Friedman test on the optimization results of HCLPSO with the same *NP* but different values of β. That is, the Friedman test is conducted within each part of this table. The second averaged rank represented as "*Rank*2" is obtained by the Friedman test on the optimization results of all HCLPSO with distinct combinations of *NP* and β. That is, the Friedman test is conducted on the whole table.

Taking a close observation on Table S3, we get the following findings:

1) In view of "*Rank*1", it is interesting to find that except for $NP = 200$, HCLPSO with different *NP* settings achieve the lowest rank when $\beta = 0.95$. This indicates that HCLPSO with $\beta = 0.95$ achieves the best overall optimization performance on the entire CEC2010 suite when *NP* varies from 300 to 700. Such a phenomenon is further demonstrated by that under different settings of *NP* (from 300 to 700), HCLPSO with $\beta = 0.95$ obtains the best results on most LSOPs.
2) With respect to "*Rank*2", it is found that HCLPSO with $NP = 500$ and $\beta = 0.95$ acquires the lowest averaged rank (3.75) among all combinations of *NP* and β. Besides, such a rank is far lower than the ones of the other combinations. This signifies that HCLPSO with $NP = 500$ and $\beta = 0.95$ attains the best overall optimization performance on the whole CEC2010 LSOP suite and presents significant superiority to HCLPSO with the other settings of *NP* and β.

Based on the above findings, $NP = 500$ along with $\beta = 0.95$ is utilized to configure HCLPSO to solve 1000-*D* LSOPs in the subsequent experiments.

4.2. Comparison with state-of-the-art large-scale approaches

This subsection performs comparative experiments to verify the optimization effectiveness and efficiency of HCLPSO by comparing it with the eight chosen large-scale methods on the two commonly utilized 1000-*D* LSOP suites. For the sake of fairness, in the experiments, we directly adopt the recommended parameter settings in the associated papers for all compared approaches, because they were also tested on the CEC2010 and the CEC2013 LSOP suites.

Table S4 and Table S5 in the Supplementary material display the detailed comparison results between HCLPSO and the eight compared methods on the CEC2010 suite and the CEC2013 suite, respectively. For better observation, Table 1 summarizes the statistical comparison results derived from the Wilcoxon rank-sum test and the Friedman test. Specifically, in this table, "*w/t/l*" means that HCLPSO significantly wins on *w* LSOPs, ties on *t* LSOPs, and loses on *l* LSOPs in competition with the associated compared methods on all LSOPs in one benchmark set. "*Rank*" denotes the average rank of each algorithm over all LSOPs in one benchmark suite.

From Table 1 and Table S4, the following findings are achieved on the twenty 1000-*D* CEC2010 LSOPs:

Table 1
Optimization performance comparison between HCLPSO and the compared large-scale optimizers on the CEC2010 and CEC2013 LSOPs.

Benchmark Set	Function Property	Index	HCLPSO	TPLSO	LLSO	SPLSO	SLPSO	CSO	DECC-DG2	DECC-RDG	DECC-RDG2
CEC2010 1000D	Fully Separable Unimodal	*w/t/l*	–	1/0/0	0/0/1	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	2/0/0	2/0/0	2/0/0
	Partially Separable Unimodal	*w/t/l*	–	1/1/4	2/1/3	6/0/0	1/1/4	6/0/0	5/0/1	4/0/2	4/0/2
	Partially Separable Multimodal	*w/t/l*	–	8/0/1	7/1/1	6/2/1	8/0/1	6/2/1	7/1/1	7/0/2	7/0/2
	Fully Non-Separable Unimodal	*w/t/l*	–	1/0/0	0/0/1	1/0/0	1/0/0	1/0/0	0/0/1	0/0/1	0/0/1
	Fully Non-Separable Multimodal	*w/t/l*	–	1/0/0	1/0/0	0/0/1	1/0/0	0/0/1	1/0/0	1/0/0	1/0/0
	Overall	*w/t/l*	–	14/1/5	11/2/7	15/2/3	14/1/5	16/2/2	16/1/3	15/0/5	15/0/5
	Overall	*Rank*	**3.00**	4.60	3.95	4.40	5.80	6.20	6.90	5.10	5.05
CEC2013 1000D	Fully Separable Unimodal	*w/t/l*	–	1/0/0	0/0/1	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	1/1/0	1/0/1	1/1/0	1/0/1	1/0/1	1/0/1
	Partially Separable Unimodal	*w/t/l*	–	0/0/3	0/2/1	0/3/0	1/0/2	2/1/0	2/0/1	2/0/1	2/0/1
	Partially Separable Multimodal	*w/t/l*	–	3/0/2	3/2/0	3/2/0	1/4/0	3/1/1	5/0/0	5/0/0	5/0/0
	Overlapping Unimodal	*w/t/l*	–	0/0/2	0/1/1	2/0/0	0/1/1	2/0/0	2/0/0	2/0/0	2/0/0
	Overlapping Multimodal	*w/t/l*	–	1/0/0	1/0/0	0/0/1	1/0/0	0/0/1	1/0/0	1/0/0	1/0/0
	Fully Non-Separable Unimodal	*w/t/l*	–	1/0/0	0/0/1	1/0/0	1/0/0	1/0/0	1/0/0	0/1/0	0/1/0
	Overall	*w/t/l*	–	7/1/7	5/6/4	8/6/1	6/5/4	9/4/2	13/0/2	12/1/2	12/1/2
	Overall	*Rank*	**3.15**	3.53	3.93	4.60	4.13	5.33	7.40	6.93	6.00

1) From the last row of the first part in Table 1, HCLPSO wins the first place among all algorithms in view of the averaged rank. Besides, it is further discovered that the rank value of HCLPSO is far smaller than the ones of the compared methods. These observations imply that HCLPSO attains the best overall optimization performance on the entire CEC2010 test suite and presents great overall dominance to the compared approaches.
2) From the second to last row in the first part of Table 1, HCLPSO obtains significantly better optimization results than the compared large-scale optimizers on more than 11 LSOPs, while it only presents inferior performance to them on no more than 7 LSOPs. Specifically, compared with TPLSO, SPLSO, SLPSO, and CSO, HCLPSO shows significantly superior performance to them on 14, 15, 14, and 16 LSOPs, respectively, while it only displays inferior performance to them on 5, 3, 5, and 2 LSOPs, respectively. These comparisons demonstrate that HCLPSO is much more effective than the compared state-of-the-art holistic large-scale PSO optimizers. In comparison with DECC-DG2, DECC-RDG2, and DECC-RDG, HCLPSO presents much better performance than them on 16, 15, and 15 LSOPs. This demonstrates that HCLPSO is also more effective than the compared state-of-the-art CC based large-scale optimizers in solving the 1000-D CEC2010 benchmark LSOPs.
3) Taking a closer observation on the first part of Table 1, we discover that on the 3 fully separable LSOPs ($F1 \sim F3$), except for LLSO and SPLSO, HCLPSO shows much better optimization results than the rest 6 compared methods on all these 3 LSOPs. Compared with LLSO and SPLSO, HCLPSO presents significant dominance to them on 1 and 2 LSOPs, respectively. On the 15 partially separable LSOPs ($F4 \sim F18$), HCLPSO shows much better optimization results than SPLSO, CSO and DECC-DG2 all on 12 LSOPs, and displays significant superiority to TPLSO, LLSO and SLPSO all on 9 LSOPs. Competed with DECC-RDG and DECC-RDG2, HCLPSO presents significantly better results both on 11 LSOPs. On the 2 fully non-separable LSOPs ($F19$ and $F20$), HCLPSO shows significantly better optimization results than TPLSO and SLPSO on the two LSOPs, while it exhibits very competitive optimization performance with the other 6 compared methods.
4) In short, HCLPSO attains the best overall optimization performance on the entire CEC2010 LSOP suite and presents significant dominance to the eight compared state-of-the-art approaches for LSOPs. Particularly, it is found that HCLPSO is much more effective in solving partially separable LSOPs, which frequently emerge in real-world applications.

From Table 1 and Table S5, we discover the following outcomes on the fifteen 1000-D CEC2013 LSOPs:

1) From the viewpoint of the averaged rank in the second part of Table 1, we find that the rank value (3.15) of HCLPSO is the smallest. This observation indicates that HCLPSO obtains the best overall optimization performance on the whole 1000-D CEC2013 LSOP suite. In particular, except for TPLSO and LLSO, the rank value of HCLPSO is much smaller than the ones of the other 6 compared methods. This reveals that HCLPSO shows significantly better overall optimization performance than the other 6 compared methods in solving such difficult LSOPs.
2) In view of "$w/t/l$" in the second part of Table 1, it is discovered that except for TPLSO, LLSO and SLPSO, HCLPSO is much superior to the other 5 compared approaches on at least 8 LSOPs, while it is only outperformed by them on at most 2 LSOPs. In particular, in comparison with DECC-DG2, DECC-RDG, and DECC-RDG2, HCLPSO significantly defeats them on at least 12 LSOPs and only presents inferiority to them on 2 LSOPs. Competed with TPLSO, LLSO, and SLPSO, HCLPSO presents highly competitive optimization performance with them.
3) Taking a closer observation on the second part of Table 1, we find that on the 3 fully separable LSOPs ($F1 \sim F3$), HCLPSO obtains competitive performance with LLSO, while it outperforms the other 7 compared algorithms all on 2 LSOPs. On the 8 partially separable LSOPs ($F4 \sim F11$), HCLPSO shows significant superiority to DECC-DG2, DECC-RDG, and DECC-RDG2 all on 7 LSOPs. In particular, it shows no worse performance than LLSO, SPLSO, SLPSO, and CSO on 7, 8, 6, and 7 LSOPs, respectively. On the 3 overlapping LSOPs ($F12 \sim F14$), HCLPSO is significantly better than DECC-DG2, DECC-RDG, and DECC-RDG2 on the three LSOPs. On the only 1 completely non-separable LSOP ($F15$), except for LLSO, HCLPSO outperforms 5 compared algorithms and achieves competitive performance with the other 2 compared methods on this LSOP.
4) In summary, on the difficult CEC2013 LSOP suite, HCLPSO presents significant dominance over 6 compared algorithms, while it achieves highly competitive performance with TPLSO and LLSO. In particular, it is found that HCLPSO shows very promising performance on partially separable LSOPs, overlapping LSOPs and completely non-separable LSOPs, which are very complicated

Table 2
Wilcoxon signed rank test results between HCLPSO and each compared algorithm on the 1000-D CEC2010 and CEC2013 benchmark sets.

Comparison	CEC2010-1000D			CEC2013-1000D		
	R+	R-	P-Value	R+	R-	P-Value
HCLPSO versus TPLSO	112	98	0.7938	38	82	0.2293
HCLPSO versus LLSO	112	98	0.7938	60	60	1.0000
HCLPSO versus SPLSO	180	30	0.0051	94	26	0.0554
HCLPSO versus SLPSO	112	98	0.7938	63	57	0.8904
HCLPSO versus CSO	188	22	0.0019	104	16	0.0103
HCLPSO versus DECC-DG2	163	47	0.0304	105	15	0.0084
HCLPSO versus DECC-RDG	144	66	0.1454	105	15	0.0084
HCLPSO versus DECC-RDG2	143	67	0.1560	99	21	0.0256

and considerably difficult to optimize. The good performance of HCLPSO on the 1000-D CEC2013 LSOPs demonstrates that it is capable of tackling complicated LSOPs.

Except for the Wilcoxon rank-sum test and the Friedman test, we also conduct the Wilcoxon signed rank test to compare the optimization performance of HCLPSO with the 8 compared methods on the 1000-D CEC2010 and the 1000-D CEC2013 benchmark suites. Table 2 shows the detailed comparison results. From this table, we find that: 1) On the CEC2010 benchmark suite, HCLPSO attains much higher “R+” than “R-” when it is competed with the 8 compared methods. Besides, on this benchmark set, HCLPSO shows significant difference from SPLSO, CSO, and DECC-DG2 at $\alpha = 0.05$, and presents significant difference from DECC-RDG and DECC-

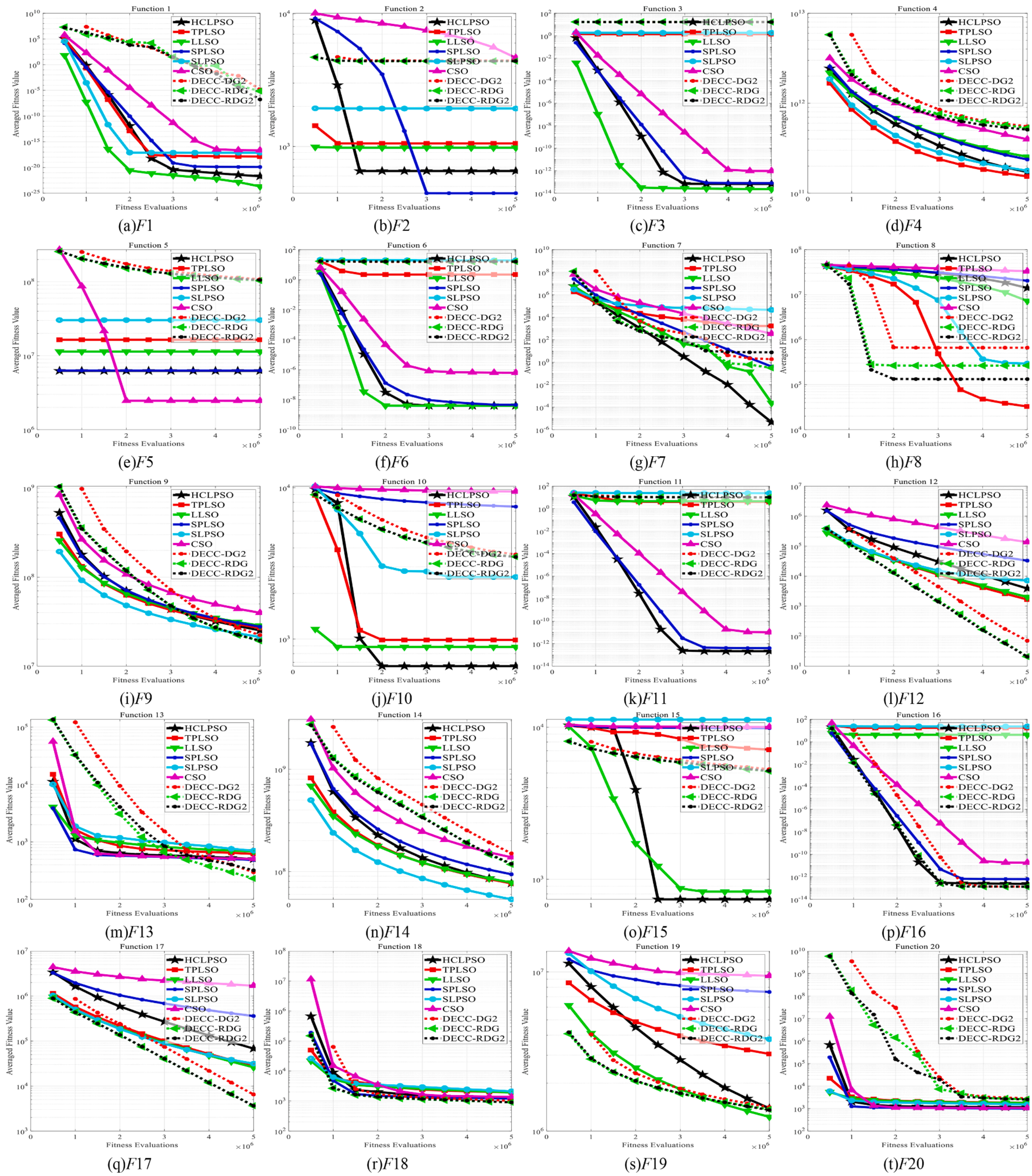

Fig. 1. Convergence behaviors of HCLPSO and the compared large-scale optimizers on the 1000-D CEC2010 LSOPs.

RDG2 at α = 0.2. 2) On the CEC2013 benchmark suite, HCLPSO consistently achieves higher "R+" than "R-" except for competing with TPLSO and LLSO. In particular, HCLPSO shows significant difference from CSO, DECC-DG2, DECC-RDG, and DECC-RDG2 at α = 0.05, and presents significant difference from TPLSO and SPLSO at α = 0.25.

On the whole, based on the above experimental results on the two LSOP benchmark sets, we find that HCLPSO is effective to tackle LSOPs and shows much superior optimization performance to most compared state-of-the-art large-scale approaches. Such superiority of HCLPSO mainly benefits from the designed heterogeneous cognitive learning strategies, which treat different kinds of particles differently and update them with different learning mechanisms. With the devised two learning schemes for inferior particles and superior particles, HCLPSO could let inferior particles explore the search space to locate optimal regions, and let superior particles exploit the optimal zones where they locate to find high-quality solutions. In this way, HCLPSO compromises search intensification and diversification during the evolution of particles to traverse the immense search space. In addition, the devised adaptive swarm partition strategy dynamically reduces the number of superior particles, leading to that HCLPSO gradually switches from exploring the search space to exploiting the found optimal regions. With this technique, HCLPSO further compromises exploration and exploitation at the swarm level. The cohesive collaboration between the heterogeneous learning schemes and the adaptive swarm partition mechanism endows HCLPSO with good ability in compromising exploration of the search space and exploitation of found optimal regions to traverse the vast search space, which makes great contributions to its good optimization performance.

The above comparative experiments have validated the optimization effectiveness of HCLPSO. To further verify the optimization efficiency of HCLPSO, experiments are further carried out on the 1000-D CEC2010 and the 1000-D CEC2013 LSOP suites to compare the convergence behaviors of HCLPSO the 8 compared large-scale methods. It deserves noting that the maximum number of fitness evaluations is 5000 × D in this experiment. Figs. 1 and 2 show the convergence behavior of HCLPSO and the 8 compared large-scale

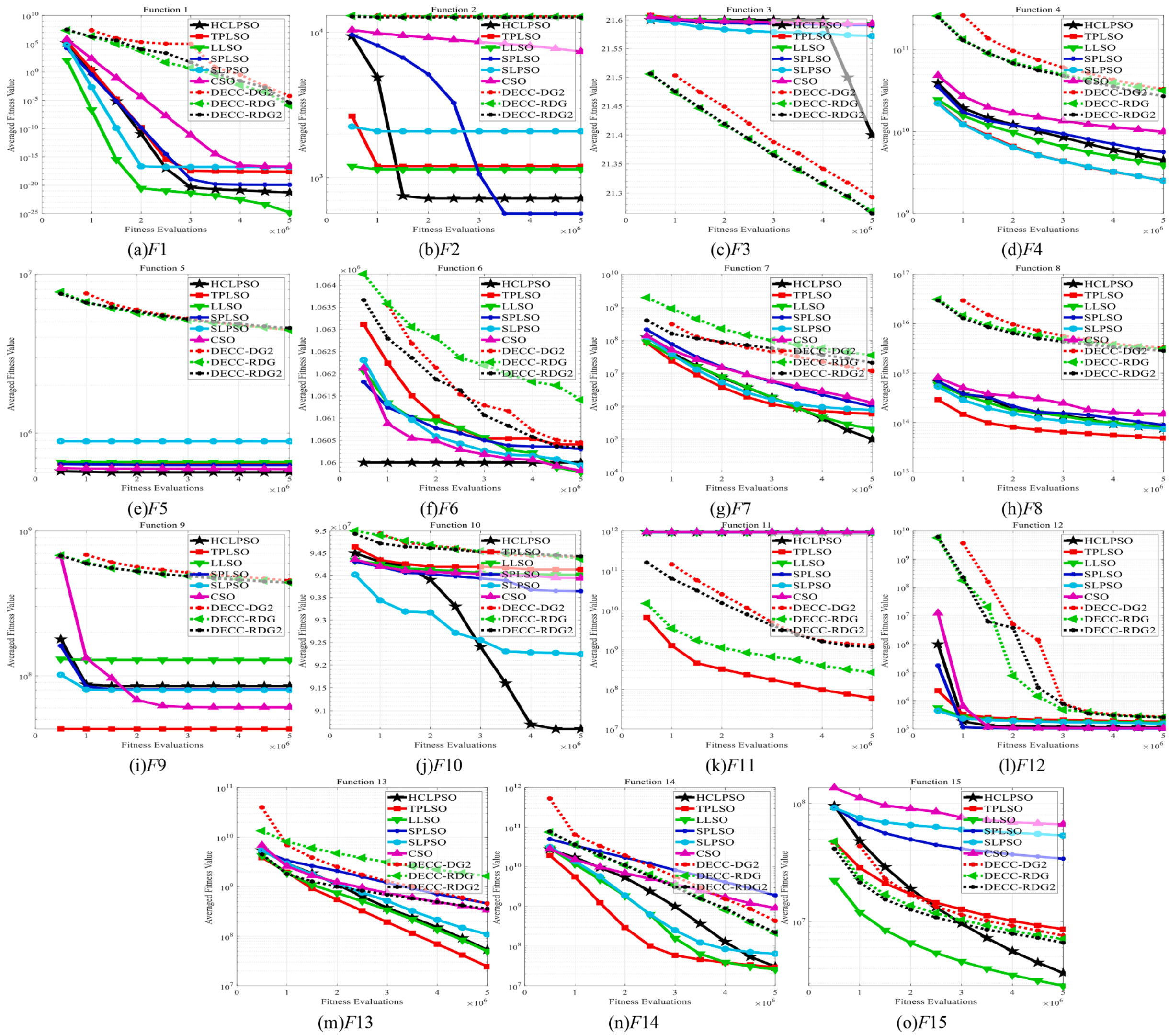

Fig. 2. Convergence behaviors of HCLPSO and the compared large-scale optimizers on the 1000-D CEC2013 LSOPs.

optimizers on the CEC2010 and the CEC2013 LSOP suites, respectively.

Observing Fig. 1, we get the following discoveries on the twenty 1000-*D* CEC2010 LSOPs:

1) At the first glimpse, it is observed that HCLPSO performs the best among all optimizers concerning the convergence speed and the solution accuracy on 4 LSOPs (*F*7, *F*10, *F*11, and *F*15).
2) Deep investigation shows that HCLPSO finds better solutions along with faster convergence speed than 7 compared approaches on 4 LSOPs (*F*1, *F*2, *F*3, and *F*6) and presents slightly worse performance than only 1 compared algorithm on these problems.
3) On *F*4, *F*5, *F*16, and *F*20, HCLPSO is much better than 6 compared approaches regarding the convergence speed and the solution quality and only displays inferiority to 2 compared algorithms.
4) On the other 9 LSOPs, HCLPSO shows better performance than at least 2 compared algorithms regarding the convergence speed and the solution quality.

Observing Fig. 2, we draw the following similar conclusions on the fifteen 1000-*D* CEC2013 LSOPs:

1) At the first look, it is found that HCLPSO obtains the best performance among all methods on 3 LSOPs (*F*5, *F*7, and *F*10) from the perspective of the solution accuracy and the convergence speed.
2) In-depth observation shows that both the convergence speed and the solution quality of HCLPSO are much superior to those of 7 compared methods on 3 LSOPs (*F*1, *F*2, and *F*15) and are only worse than those of 1 compared method on the three LSOPs.
3) Further, it is found that on *F*8, *F*12, *F*13, and *F*14, HCLPSO performs much better than 6 compared approaches and displays inferiority to only 2 compared methods concerning both the solution accuracy and the convergence speed.
4) On the other 5 LSOPs, HCLPSO presents superiority to at least 4 compared approaches regarding the solution quality and the convergence speed.

To summarize, the above comparison results verify that HCLPSO achieves higher solution quality along with faster convergence than most compared large-scale algorithms on most benchmark LSOPs in the two test suites. This demonstrates that HCLPSO is efficient to cope with LSOPs. Such an advantageous property of HCLPSO also benefits from the cooperation between the devised heterogeneous learning schemes and the designed dynamic swarm partition technique. Particularly, these two techniques afford a good compromise between exploration and exploitation at the swarm level and the particle level for HCLPSO. As a result, HCLPSO is capable of searching the vast solution space with intensification and diversification advisably to find as accurate solutions as possible.

4.3. Deep investigation on HCLPSO

As illustrated above, the good performance of HCLPSO in solving LSOPs mainly benefits from the devised heterogeneous learning schemes and the designed dynamic swarm partition technique. To verify this, we execute investigation experiments on the CEC2010 LSOP suite to validate the usefulness of the devised two techniques.

4.3.1. Effectiveness of the devised heterogeneous learning strategies

This subsection aims to investigate the usefulness of the devised two learning strategies for inferior particles and superior particles. To this end, we develop six different versions of HCLPSO to make comparisons with the original HCLPSO.

Firstly, as for the devised RECL for inferior particles as elucidated in Section 3.1.1, we design a non-linear weight function to calculate the selection probability of superior particles. To validate the effectiveness of this non-linear calculation method, we develop three other selection approaches for choosing superior particles to direct the evolution of the inferior ones. The first one is also based on the roulette wheel selection strategy, but the selection probabilities of superior particles are computed according to their fitness as follows:

$$f^{`}\left(\boldsymbol{x}_{SP,i}\right) = \frac{f_{\max} - f\left(\boldsymbol{x}_{SP,i}\right) + \xi}{f_{\max} - f_{\min} + \xi} \tag{11}$$

$$P_i = \frac{f^{`}\left(\boldsymbol{x}_{SP,i}\right)}{\sum_{j=1}^{NSP} f^{`}\left(\boldsymbol{x}_{SP,j}\right)} \tag{12}$$

where $f_{\min}$ and $f_{\max}$ represent the minimum and the maximum fitness values of superior particles, respectively. $f(\boldsymbol{x}_{SP,i})$ is the fitness of the *i*th superior particle. ξ denotes a tiny real value adopted to prevent the denominator from being zero. *NSP* denotes the number of superior particles. $f^{`}(x_{SP,i})$ represents the normalized fitness value of the *i*th superior particle, and P_i is its selection probability.

Using this selection method to replace the one in HCLPSO, we develop a version of HCLPSO, which is denoted as "HCLPSO-FS" meaning that it uses the fitness of superior particles to compute their selection probabilities.

The second version of HCLPSO is also based on the roulette wheel selection strategy, but the selection probabilities of superior particles are computed directly based on their rankings in the following way instead of using the non-linear weight function in the original HCLPSO:

$$P_i = \frac{NSP - rank(i) + 1}{\sum_{j=1}^{NSP} j} \tag{13}$$

where P_i denotes the selection probability of the ith superior particle, and $rank(i)$ is its ranking after all superior particles are ranked from the smallest to the largest regarding their fitness, and NSP is the number of superior particles.

Replacing the selection method in the original HCLPSO with this selection scheme, we develop another version of HCLPSO, which is denoted as “HCLPSO-RS” meaning that it directly employs the rankings of superior particles to compute their selection probabilities.

The third version of HCLPSO is to randomly choose a superior particle from ***SP*** with the uniform distribution as the leading exemplar of each inferior particle. Unlike the non-linear selection method in HCLPSO and the above two developed selection approaches, this selection method assumes that all superior particles have the same chance to be chosen as the guiding exemplars of inferior particles. We denote this version of HCLPSO as “HCLPSO-US”.

Secondly, as for the devised SDCL learning scheme for superior particles, to investigate its usefulness, we develop another three versions of HCLPSO. First, we remove this learning strategy from HCLPSO and leave superior particles to be not updated in each generation. We denote this version of HCLPSO as “HCLPSO-WSDCL”. Second, in SDCL, a superior particle is evolved only when the randomly chosen guiding exemplar from ***SP*** is no worse than itself. Such a stochastic competition mechanism affords chances for relatively poor superior particles to survive, which is conducive to the swarm diversity promotion. To validate this, we develop a dominant cognitive learning (DCL) scheme by removing the stochastic competition. To be specific, for each superior particle, we randomly select a guiding exemplar from those which are ranked ahead of it to guide its evolution. In this way, (*NSP*-1) superior particles are updated in each generation except for the first ranked superior particle (because there are no better superior particles to direct its update). We denote this version of HCLPSO as “HCLPSO-DCL”. At last, we directly utilize the selection method in RECL devised for inferior particles to choose leading exemplars to guide the evolution of superior particles. We denote this version of HCLPSO as “HCLPSO-RECL”.

After the above preparations, we conduct experiments on the 1000-*D* CEC2010 test suite to compare HCLPSO with the above developed six versions. Table 3 presents the optimization results of distinct HCLPSO variants on the 20 CEC2010 LSOPs. The best optimization results are bolded, and the averaged ranks of all versions of HCLPSO are shown in the last row.

From Table 3, we attain the following discoveries:

1) Regarding the averaged ranks displayed in the last row, the original HCLPSO attains the smallest rank among all versions and such a rank value is far smaller than the ones of the developed six versions. This signifies that the original HCLPSO acquires the best overall optimization performance on the entire CEC2010 LSOP suite and its optimization performance is significantly superior to those of the developed six versions. This demonstrates the effectiveness of the devised two heterogeneous learning strategies, whose cohesive cooperation results in the good optimization performance of HCLPSO.
2) Compared with HCLPSO-FS, HCLPSO-RS, and HCLPSO-US, HCLSO obtains much better performance. This indicates that the non-linear selection method devised in RECL is much more effective than the three compared selection methods. This is mainly because the devised non-linear weight function treats different superior particles differently and lets superior particles with better fitness have exponentially higher selection probabilities. In this way, the devised selection method affords a potentially better balance between exploration and exploitation for inferior particles to locate promising regions. Therefore, the effectiveness of the devised RECL for inferior particles is verified.

Table 3
Optimization performance of different versions of HCLPSO on the 1000-*D* CEC2010 LSOPs.

F	HCLPSO	HCLPSO-FS	HCLPSO-RS	HCLPSO-US	HCLPSO-WSDCL	HCLPSO-DCL	HCLPSO-RECL
*F*1	3.74E−21	7.81E−17	8.77E−16	1.17E−10	**1.57E−21**	7.22E−21	2.56E−21
*F*2	6.54E+02	**4.89E+02**	1.29E+03	8.07E+03	1.79E+03	9.34E+02	1.29E+03
*F*3	**6.91E−14**	1.08E−11	4.13E−11	1.59E−08	7.34E−14	1.07E−13	6.13E−02
*F*4	3.36E+11	5.09E+11	5.56E+11	6.84E+11	**2.37E+11**	4.43E+11	3.30E+11
*F*5	6.24E+06	3.38E+06	**3.02E+06**	1.24E+07	6.22E+07	2.06E+07	1.42E+07
*F*6	**4.00E−09**	6.03E−09	7.64E−09	9.80E−08	1.98E+01	2.50E−01	2.71E+00
*F*7	3.04E+00	1.36E+03	1.67E+03	8.37E+03	**7.14E−02**	1.02E+01	1.12E+01
*F*8	3.03E+07	3.57E+07	3.67E+07	3.85E+07	**1.07E+07**	2.70E+07	1.42E+07
*F*9	4.37E+07	5.89E+07	5.87E+07	7.29E+07	**3.93E+07**	5.72E+07	5.10E+07
*F*10	**6.61E+02**	7.22E+03	8.33E+03	9.59E+03	1.93E+03	8.44E+03	1.23E+03
*F*11	**2.42E−13**	1.42E−10	5.15E−10	1.83E−07	1.97E+01	1.11E−01	1.76E+01
*F*12	3.05E+04	1.28E+05	1.41E+05	6.49E+05	**9.04E+03**	4.29E+04	1.53E+04
*F*13	5.84E+02	6.69E+02	5.59E+02	5.74E+02	6.40E+02	**5.99E+02**	8.24E+02
*F*14	1.36E+08	2.00E+08	2.07E+08	2.96E+08	**1.14E+08**	1.62E+08	1.59E+08
*F*15	**7.38E+02**	9.95E+03	9.97E+03	1.01E+04	1.96E+03	1.03E+04	1.18E+03
*F*16	**2.97E−13**	2.27E−10	9.30E−10	2.93E−02	9.78E+00	9.26E−01	8.94E+00
*F*17	2.69E+05	1.02E+06	1.25E+06	2.53E+06	**9.64E+04**	3.44E+05	1.19E+05
*F*18	1.51E+03	**1.43E+03**	1.57E+03	1.52E+03	1.90E+03	1.59E+03	2.58E+03
*F*19	2.80E+06	9.21E+06	8.72E+06	1.08E+07	**1.35E+06**	7.81E+06	1.46E+06
*F*20	1.20E+03	**1.03E+03**	1.07E+03	1.04E+03	1.77E+03	1.17E+03	2.14E+03
Rank	**2.30**	3.80	4.45	5.60	3.40	4.35	4.10

3) Compared with HCLPSO-WSDSL, HCLPSO-DCL, and HCLPSO-RECL, HCLPSO attains far better optimization performance. This proves the effectiveness of the devised SDCL for superior particles. In particular, the superiority of HCLPSO over HCLPSO-WSDCL indicates that SDCL is very helpful for HCLPSO to achieve good optimization performance. The dominance of HCLPSO to HCLPSO-DCL demonstrates the usefulness of the stochastic competition in the SDCL strategy, which affords chances for relatively poor superior particles to survive. The better performance of HCLPSO than HCLPSO-RECL demonstrates the usefulness of treating the two kinds of particles differently.

Comprehensively speaking, the above comparative experiments have testified the effectiveness of the devised RECL strategy for inferior particles and the designed SDCL strategy for superior particles and the effectiveness of the cooperation between these two learning schemes.

In RECL, as shown in Eq. (3), there is a key parameter (namely σ) in the devised non-linear weight function. It directly affects the selection probabilities of superior particles. To investigate its influence on the optimization performance of HCLPSO, we conduct experiments on the 1000-D CEC2010 benchmark suite by varying σ from 0.05 to 0.9. Table S6 in the Supplementary material presents the optimization results. In this table, the best results are bolded and the average rank of each configuration obtained from the Friedman test is listed in the last row.

From Table S6, it is found that: 1) In view of the average rank, $\sigma = 0.1$ helps HCLPSO achieve the smallest rank value and thus it helps HCLPSO obtain the best overall optimization performance on the whole CEC2010 benchmark suite. 2) In particular, a large σ is not preferable for HCLPSO because when $\sigma > 0.3$, not only the rank value of HCLPSO is very large, but also HCLPSO achieves the best results on at most 1 LSOP. 3) A small σ is preferred because a small σ affords large weights for the top ranked superior particles, leading to that the selection of guiding exemplars for inferior particles biases to choosing the top ranked superior particles. As a result, with RECL, it is likely that inferior particles move toward optimal regions fast.

Comprehensively, based on the above investigation experiments, we set $\sigma = 0.1$ for HCLPSO in this paper for all experiments.

4.3.2. Effectiveness of the devised adaptive swarm partition mechanism

In this subsection, we investigate the influence of the devised adaptive swarm partition mechanism elucidated in Section 3.2. Specifically, the adaptive strategy dynamically adjusts the value of *SPR* to realize the dynamic partition of the swarm into superior particles and inferior particles. To validate its usefulness, we fix *SPR* with values varying from 0.20 to 0.90 instead of the adaptive mechanism. Next, we perform comparative experiments on the 1000-D CEC2010 LSOPs to compare HCLPSO with the adaptive strategy and the ones with distinct fixed *SPR*. Table 4 displays the comparison results with the best optimization results highlighted in bold. Furthermore, the averaged ranks of different HCLPSO variants are presented in the last row.

Observing Table 4, we discover the following findings:

1) Concerning the averaged rank, it is observed that HCLPSO with the adaptive strategy acquires the smallest averaged rank and its rank value is far lower than the ones of the other HCLPSO variants. This indicates that HCLPSO with the adaptive scheme attains the best overall optimization results on the whole CEC2010 test suite and its optimization performance is significantly superior to HCLPSO with the fixed setting of *SPR*. This validates the effectiveness and usefulness of the adaptive swarm partition method.
2) Furthermore, we discover that on different LSOPs, the optimal setting of *SPR* is distinct for HCLPSO to get the best optimization results. This implies that HCLPSO is sensitive to *SPR*. However, with the adaptive strategy, such sensitivity can be alleviated.

Table 4
Optimization performance of HCLPSO with the adaptive adjustment of *SPR* and the ones with distinct *SPR* settings on the 1000-D CEC2010 LSOPs.

F	*SPR* Adaptive	*SPR* = 0.20	*SPR* = 0.30	*SPR* = 0.40	*SPR* = 0.50	*SPR* = 0.60	*SPR* = 0.70	*SPR* = 0.80	*SPR* = 0.90
*F*1	3.74E−21	1.13E+02	2.30E−21	**1.68E−21**	2.09E−21	2.70E−21	7.52E−21	1.22E−16	1.47E−10
*F*2	6.54E+02	3.37E+03	2.46E+03	1.89E+03	1.43E+03	1.04E+03	7.47E+02	**5.43E+02**	7.51E+03
*F*3	6.91E−14	4.61E+00	1.66E+00	1.21E−01	7.39E−14	6.68E−14	**6.42E−14**	1.16E−11	1.54E−08
*F*4	**3.36E+11**	3.87E+11	3.45E+11	3.59E+11	3.93E+11	4.66E+11	4.49E+11	5.24E+11	6.10E+11
*F*5	6.24E+06	3.91E+07	2.70E+07	2.12E+07	1.54E+07	1.20E+07	9.32E+06	6.80E+06	**3.56E+06**
*F*6	**4.00E−09**	1.99E+01	1.98E+01	1.96E+01	1.63E−01	4.00E−09	4.13E−09	7.39E−09	1.15E−07
*F*7	3.04E+00	2.66E+05	5.00E+03	**7.41E−01**	1.32E+00	8.57E+00	5.66E+01	5.02E+02	4.60E+03
*F*8	3.03E+07	1.85E+07	**1.34E+07**	2.27E+07	2.65E+07	3.05E+07	3.52E+07	3.60E+07	3.82E+07
*F*9	4.37E+07	7.31E+07	4.75E+07	**4.36E+07**	4.51E+07	4.56E+07	5.05E+07	5.68E+07	7.32E+07
*F*10	**6.61E+02**	3.55E+03	2.59E+03	1.93E+03	1.45E+03	1.05E+03	6.78E+02	7.66E+03	9.55E+03
*F*11	2.42E−13	5.52E+01	2.17E+01	1.98E+01	1.29E+01	**2.04E−13**	4.98E−13	2.54E−10	2.32E−07
*F*12	3.05E+04	6.73E+04	1.42E+04	**1.37E+04**	1.73E+04	2.69E+04	4.67E+04	1.01E+05	5.86E+05
*F*13	5.84E+02	2.65E+03	9.22E+02	7.60E+02	7.73E+02	6.62E+02	6.89E+02	6.64E+02	**5.80E+02**
*F*14	1.36E+08	2.49E+08	1.42E+08	**1.27E+08**	1.30E+08	1.37E+08	1.56E+08	1.95E+08	2.92E+08
*F*15	**7.38E+02**	3.77E+03	2.78E+03	2.02E+03	1.46E+03	9.51E+02	9.80E+03	9.97E+03	1.01E+04
*F*16	**2.97E−13**	1.73E+02	5.19E+01	1.23E+01	2.73E+00	4.86E−01	5.18E−13	2.93E−02	3.72E−07
*F*17	2.69E+05	2.70E+05	**1.13E+05**	1.17E+05	1.51E+05	2.23E+05	3.99E+05	9.69E+05	2.47E+06
*F*18	**1.51E+03**	7.52E+03	2.64E+03	2.40E+03	2.30E+03	2.00E+03	2.03E+03	1.95E+03	1.75E+03
*F*19	2.80E+06	1.75E+06	**1.28E+06**	1.30E+06	1.59E+06	2.57E+06	5.49E+06	8.10E+06	1.01E+07
*F*20	1.20E+03	3.94E+03	2.18E+03	1.92E+03	1.75E+03	1.50E+03	1.30E+03	1.10E+03	**1.08E+03**
Rank	**2.70**	7.50	5.50	4.25	4.30	3.90	4.75	5.65	6.45

Table 5
Optimization performance comparison between HCLPSO and the compared large-scale optimizers on the 200-*D*, 500-*D*, 800-*D* and 2000-*D* CEC2010 LSOPs.

Benchmark Set	Function Property	Index	HCLPSO	TPLSO	LLSO	SPLSO	SLPSO	CSO	DECC-DG2	DECC-RDG	DECC-RDG2
CEC2010 200*D*	Fully Separable Unimodal	*w/t/l*	–	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1//0/0	0/1/0	1/0/0
	Fully Separable Multimodal	*w/t/l*	–	2/0/0	1/0/1	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0
	Partially Separable Unimodal	*w/t/l*	–	1/1/4	0/3/3	5/1/0	3/1/2	6/0/0	5/0/1	5/0/1	5/0/1
	Partially Separable Multimodal	*w/t/l*	–	8/0/1	4/1/1	6/3/0	8/1/0	9/0/0	7/1/1	7/1/1	8/1/0
	Fully Non-Separable Unimodal	*w/t/l*	–	0/0/1	0/1/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0
	Fully Non-Separable Multimodal	*w/t/l*	–	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*	–	13/1/6	6/6/8	16/4/0	16/2/2	20/0/0	17/1/2	16/2/2	18/1/1
	Overall	*Rank*	**2.25**	4.00	3.00	4.25	4.25	7.00	6.15	5.65	6.30
CEC2010 500*D*	Fully Separable Unimodal	*w/t/l*	–	1/0/0	0/0/1	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	2/0/0	2/0/0	2/0/0
	Partially Separable Unimodal	*w/t/l*	–	3/0/3	1/2/3	4/0/2	3/1/2	6/0/0	5/0/1	4/0/2	4/0/2
	Partially Separable Multimodal	*w/t/l*	–	8/0/1	7/1/1	4/3/2	8/0/1	9/0/0	7/1/1	7/1/1	7/1/1
	Fully Non-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
	Fully Non-Separable Multimodal	*w/t/l*	–	1/0/0	1/0/0	0/0/1	1/0/0	0/1/0	1/0/0	1/0/0	1/0/0
	Overall	*w/t/l*	–	16/0/4	11/3/6	11/3/6	16/1/3	19/1/0	17/1/2	16/1/3	16/1/3
	Overall	*Rank*	**2.60**	5.55	3.75	3.55	6.65	6.60	6.75	4.85	4.90
CEC2010 800*D*	Fully Separable Unimodal	*w/t/l*	–	1/0/0	0/0/1	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	0/0/2	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0
	Partially Separable Unimodal	*w/t/l*	–	5/1/0	5/0/1	3/1/2	3/1/2	6/0/0	5/0/1	4/0/2	4/0/2
	Partially Separable Multimodal	*w/t/l*	–	8/0/1	5/3/1	4/4/1	8/0/1	6/2/1	6/2/1	6/0/3	6/0/3
	Fully Non-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
	Fully Non-Separable Multimodal	*w/t/l*	–	1/0/0	1/0/0	0/0/1	1/0/0	0/0/1	1/0/0	1/0/0	1/0/0
	Overall	*w/t/l*	–	18/1/1	13/3/4	9/5/6	16/1/3	16/2/2	16/2/2	15/0/5	15/0/5
	Overall	*Rank*	**2.60**	5.40	4.30	3.85	6.30	6.30	6.65	4.85	4.75
CEC2010 2000*D*	Fully Separable Unimodal	*w/t/l*	–	1/0/0	0/0/1	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	0/0/2	1/0/1	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0
	Partially Separable Unimodal	*w/t/l*	–	4/1/1	6/0/0	2/1/3	6/0/0	6/0/0	5/0/1	5/0/1	5/0/1
	Partially Separable Multimodal	*w/t/l*	–	8/0/1	2/2/5	5/3/1	9/0/0	4/2/3	8/1/0	6/0/3	6/0/3
	Fully Non-Separable Unimodal	*w/t/l*	–	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	0/0/1	0/0/1
	Fully Non-Separable Multimodal	*w/t/l*	–	1/0/0	0/0/1	0/0/1	1/0/0	0/0/1	1/0/0	1/0/0	1/0/0
	Overall	*w/t/l*	–	17/1/2	9/2/9	10/4/6	20/0/0	14/2/4	18/1/1	15/0/5	15/0/5
	Overall	*Rank*	**2.75**	4.70	3.40	3.65	7.30	5.00	7.95	5.20	5.05

Besides, with the help of the adaptive mechanism, HCLPSO acquires the best optimization results on 5 LSOPs and the difference between the results obtained by HCLPSO with the adaptive strategy and those obtained by HCLPSO with the optimal fixed *SPR* on these 5 problems is very significant.

In short, the above experiments have validated the effectiveness of the devised adaptive swarm partition strategy. In particular, such a strategy affords a potential compromise between exploitation and exploration at the swarm level. With this mechanism, HCLPSO gradually changes from exploring the search space for locating optimal areas to exploiting the found optimal zones to find accurate solutions and thus is promising to solve LSOPs.

In addition, as shown in Eq. (9), there is a parameter (namely SPR_{max}) in the devised adaptive swarm partition strategy. It controls the change range of *SPR* and thus has a great effect on the optimization performance of HCLPSO. Therefore, to investigate its influence, we conduct experiments on the 1000-*D* CEC2010 benchmark suite by varying SPR_{max} from 0.2 to 1.0. Table S7 in the Supplementary material shows the optimization results of HCLPSO with different SPR_{max}. In this table, the best results are highlighted in bold and the average ranks of all configurations of SPR_{max} obtained from the Friedman test are displayed in the last row.

From Table S7, we find that: 1) Regarding the Friedman test results, HCLPSO with $SPR_{max} = 0.9$ obtains the smallest average rank, which indicates that $SPR_{max} = 0.9$ helps HCLPSO attain the best overall optimization performance on the whole CEC2010 suite. 2) In particular, it is interesting to find that a large SPR_{max} is preferred because when SPR_{max} increases from 0.2 to 0.9, the rank value of HCLPSO becomes smaller and smaller. This is because it affords more diversified exemplar selection when SPR_{max} starts from a large value, which is conductive to the great promotion of search diversity.

Comprehensively speaking, based on the above investigation experiments, we set $SPR_{max} = 0.9$ for HCLPSO in this paper.

4.4. Scalability investigation of HCLPSO

In this subsection, we aim to investigate the scalability of HCLPSO to solve LSOPs with different dimensionalities. To this end, we first generates LSOPs with different dimension sizes by modifying the problem generators in the CEC2010 benchmark set. Specifically, we generate four variant sets of the CEC2010 suite by setting the dimension size as 200, 500, 800, and 2000, respectively. Subsequently, based on these four problem sets, we conduct comparative experiments to compare HCLPSO with the 8 compared large-scale algorithms. Tables S8, S9, S10, and S11 in the Supplementary material present the detailed comparison results between HCLPSO and the 8 compared methods on the 200-*D*, 500-*D*, 800-*D*, and 2000-*D* CEC2010 suites, respectively. For better observation, we summarize the statistical comparison results between HCLPSO and the 8 compared methods on the four CEC2010 suites based on the Wilcoxon rank-sum test and the Friedman test in Table 5. In addition, Table 6 displays the Wilcoxon signed rank test results between HCLPSO and each compared method on the four CEC2010 suites.

Observing Table 5, Table 6, and Table S8, we get the following findings on the 200-*D* CEC2010 suite:

1) In view of the last row of the first part in Table 5, HCLPSO gets the lowest rank value among all algorithms. Besides, its rank value is much smaller than those of the 8 compared methods. These two observations imply that HCLPSO not only achieves the best overall

Table 6
Wilcoxon signed rank test results between HCLPSO and each compared algorithm on the CEC2010 benchmark set with different settings of the dimensionality.

Comparison	CEC2010-200D			CEC2010-500D		
	R+	R−	P-Value	R+	R−	P-Value
HCLPSO versus TPLSO	100	110	0.8519	147	63	0.1169
HCLPSO versus LLSO	117	93	0.6542	109	101	0.8813
HCLPSO versus SPLSO	210	0	0.0001	135	75	0.2627
HCLPSO versus SLPSO	156	54	0.0569	155	55	0.0620
HCLPSO versus CSO	210	0	0.0001	204	6	0.0002
HCLPSO versus DECC-DG2	183	27	0.0036	180	30	0.0051
HCLPSO versus DECC-RDG	182	28	0.0040	162	48	0.0333
HCLPSO versus DECC-RDG2	197	13	0.0006	162	48	0.0333
Comparison	**CEC2010-800D**			**CEC2010-2000D**		
	R+	R−	P-Value	R+	R−	P-Value
HCLPSO versus TPLSO	174	36	0.0100	181	29	0.0045
HCLPSO versus LLSO	186	24	0.0025	150	60	0.0930
HCLPSO versus SPLSO	149	61	0.1005	126	84	0.4330
HCLPSO versus SLPSO	141	69	0.0025	210	0	0.0001
HCLPSO versus CSO	186	24	0.0025	178	32	0.0064
HCLPSO versus DECC-DG2	168	42	0.0187	198	12	0.0005
HCLPSO versus DECC-RDG	151	59	0.0859	164	46	0.0276
HCLPSO versus DECC-RDG2	151	59	0.0859	163	47	0.0304

performance on the whole 200-D CEC2010 suite among all algorithms, but also displays significant superiority to the 8 compared methods.

2) From the perspective of the second to last row in the first part of Table 5, except for LLSO, HCLPSO significantly outperforms the other 7 compared methods on at least 13 LSOPs, and only shows significant inferiority to them on at most 6 LSOPs. In particular, HCLPSO is significantly better than CSO on all twenty 200-D CEC2010 LSOPs, shows significant superiority to SPLSO and SLPSO both on 16 LSOPs, and displays much better performance than the three CC-based large-scale approaches on more than 16 problems.
3) Regarding the Wilcoxon signed rank test results shown in Table 6, HCLPSO attains much higher "R+" than "R-". Besides, it shows significant difference from SPLSO, CSO, DECC-DG2, DECC-RDG, and DECC-RDG2 at $\alpha = 0.05$, and displays significant difference from SLPSO at $\alpha = 0.06$.
4) On different kinds of LSOPs, closer observations on the first part of Table 5 reveal that on the three fully separable LSOPs ($F1 \sim F3$), except for LLSO and DECC-RDG, HCLPSO performs significantly better than the other 6 compared algorithms on all these three problems; even compared with LLSO and DECC-RDG, HCLPSO still significantly outperforms them on 2 LSOPs. On the 15 partially separable LSOPs ($F4 \sim F18$), HCLPSO attains considerably competitive performance with LLSO, shows significantly better performance than TPLSO on 9 problems, and significantly wins the competitions with the other 6 compared methods on more than 11 problems; in particular, HCLPSO is much better than CSO on all these 15 LSOPs. On the two fully non-separable LSOPs, except for TPLSO and LLSO, with which HCLPSO achieves highly competitive performance, HCLPSO presents significant dominance to the other 6 compared methods on the two problems. The above findings reveal that HCLPSO is particularly good at solving complicated LSOPs, such as fully non-separable and partially separable LSOPs.

Observing Table 5, Table 6, and Table S9, we attain the following observations on the 500-D CEC2010 suite:

1) Regarding the Friedman test results in the second part of Table 5, HCLPSO ranks the first place among all methods. Besides, its rank value is far smaller than the ones of the 8 compared methods. These comparison results reveal that on the one hand, HCLPSO attains the best overall optimization performance on the entire 500-D CEC2010 suite among all algorithms; on the other hand, its optimization performance is significantly better than those of the 8 compared methods.
2) With respect to "*w/t/l*" in the second part of Table 5, apart from LLSO and SPLSO, HCLPSO shows significant dominance to the other 6 compared methods on more than 16 LSOPs, and only displays significant inferiority to them on no more than 4 LSOPs. Even compared with LLSO and SPLSO, HCLPSO significantly dominates them on 11 problems, and only performs worse than them on 6 problems.
3) As for the Wilcoxon signed rank test results shown in Table 6, competed with the 8 compared methods, HCLPSO attains much higher "R+" than "R-" except for competing with LLSO. Besides, it shows significant difference from CSO, DECC-DG2, DECC-RDG, and DECC-RDG2 at $\alpha = 0.05$, and presents significant difference from SLPSO at $\alpha = 0.07$.
4) On different types of LSOPs, deep observations on the second part of Table 5 discover that on the three fully separable LSOPs ($F1 \sim F3$), except for LLSO and SPLSO, HCLPSO significantly outperforms the other 6 compared algorithms on all these three problems. On the 15 partially separable LSOPs ($F4 \sim F18$), HCLPSO attains much better performance than LLSO and SPLSO on 8 LSOPs, and performs significantly better than the other 6 compared methods on more than 11 problems; particularly, HCLPSO performs much better than CSO on all these 15 LSOPs. On the two fully non-separable LSOPs, except for SPLSO and CSO, with which HCLPSO attains considerably competitive performance, HCLPSO presents significant superiority to the other 6 compared methods on the two problems. The above observations demonstrate that HCLPSO is very promising for solving complicated LSOPs.

Observing Table 5, Table 6, and Table S10, we achieve the following discoveries on the 800-D CEC2010 suite:

1) Observing the last row of the third part in Table 5, we find that HCLPSO gains the smallest rank value among all algorithms. Besides, in comparison with the rank values of the 8 compared methods, the rank value of HCLPSO is much smaller. These comparisons verify that HCLPSO both gains the best overall performance on the whole 800-D CEC2010 suite among all algorithms, and performs significantly better than the 8 compared methods.
2) From the perspective of the Wilcoxon rank-sum test results, except for SPLSO, HCLPSO performs significantly better than the other 7 compared methods on at least 13 LSOPs, and only performs significantly worse than them on at most 5 LSOPs. In particular, HCLPSO obtains significantly better performance than TPLSO on 18 LSOPs, presents significant dominance to LLSO on 13 LSOPs, shows significant superiority to SLPSO, CSO, and DECC-DG2 all on 16 LSOPs, and displays much better performance than DECC-RDG and DECC-RDG2 on 15 problems.
3) In terms of the Wilcoxon signed rank test results shown in Table 6, HCLPSO gains much higher "R+" than "R-". Besides, it shows significant difference from TPLSO, LLSO, SLPSO, CSO, and DECC-DG2 at $\alpha = 0.05$, while it displays significant difference from SPLSO, DECC-RDG, and DECC-RDG2 at $\alpha = 0.1$.
4) On different classes of LSOPs, further observations on the third part of Table 5 uncover that on the three fully separable LSOPs ($F1 \sim F3$), except for LLSO and SPLSO, HCLPSO gains significantly better performance than the other 6 compared algorithms on all these three problems. On the 15 partially separable LSOPs ($F4 \sim F18$), HCLPSO attains considerably better performance than SPLSO on 7 LSOPs and fails to compete with it on only 3 LSOPs. Compared with the other 7 compared methods, HCLPSO shows significantly better performance on more than 10 problems and displays inferiority to them on at most 5 problems. On the two fully non-separable LSOPs, except for SPLSO and CSO, with which HCLPSO achieves highly competitive performance, HCLPSO

significantly dominates the other 6 compared methods on the two problems. The above discoveries demonstrate that HCLPSO is particularly promising for solving complicated LSOPs, like fully non-separable and partially separable LSOPs.

Observing Table 5, Table 6, and Table S11, we discover the following observations on the 2000-D CEC2010 suite:

1) In terms of the last row of the last part in Table 5, HCLPSO still ranks the first place among all algorithms. Besides, its rank value is still far smaller than those of the 8 compared methods. These signify that HCLPSO not only performs the best on the whole 2000-D CEC2010 suite among all algorithms, but also significantly outperforms the 8 compared methods.
2) From the perspective of "*w/t/l*", the optimization performance of HCLPSO is significantly better than SPLSO on 10 problems, is competitive to LLSO, but is significantly better than the other 6 compared methods on at least 14 LSOPs. In particular, HCLPSO is significantly better than SLPSO on all twenty 2000-D CEC2010 LSOPs.
3) With respect to the Wilcoxon signed rank test results shown in Table 6, HCLPSO still attains much higher "R+" than "R-". Besides, it shows significant difference from TPLSO, SLPSO, CSO, DECC-DG2, DECC-RDG, and DECC-RDG2 at $\alpha = 0.05$, while it displays significant difference from LLSO at $\alpha = 0.1$.
4) On different categories of LSOPs, closer observations on the last part of Table 5 reveal that on the three fully separable LSOPs ($F1 \sim F3$), except for LLSO and SPLSO, HCLPSO performs significantly better than the other 6 compared algorithms on all these three problems. On the 15 partially separable LSOPs ($F4 \sim F18$), HCLPSO attains better performance than LLSO and SPLSO on 8 and 7 problems, respectively, while it shows significantly better performance than the other 6 compared methods on more than 10 problems; in particular, HCLPSO is much better than SLPSO on all these 15 LSOPs. On the two fully non-separable LSOPs, HCLPSO presents significant dominance to TPLSO, SLPSO, and DECC-DG2 on the two problems, while it achieves highly competitive performance with the other 5 compared methods. The above findings reveal that HCLPSO is not only promising for solving simple LSOPs, but also suitable for solving complicated LSOPs.

Overall, based on the above experiments, it is discovered that HCLPSO consistently achieves the lowest average rank among all algorithms on these four CEC2010 suites. Besides, HCLPSO performs much better than most of the compared methods on most LSOPs in the four LSOP sets. These comprehensively demonstrate that HCLPSO preserves a good scalability to solve LSOPs with different dimensionalities. Such a good property of HCLPSO also mainly benefits from the devised heterogeneous learning strategies and the designed adaptive swarm partition mechanism. The cooperation between the two techniques endows HCLPSO a good capability in balancing exploration of the vast search space and exploitation of the found optimal regions, which contributes to its good optimization performance in solving LSOPs, especially the complicated ones.

5. Conclusion

In this article, we have devised a heterogeneous cognitive learning particle swarm optimization (HCLPSO) approach to cope with large-scale optimization problems (LSOPs). Specifically, particles in the swarm are treated differently by dividing them into two groups, namely superior particles (***SP***) and inferior particles (***IP***). Then, for inferior particles, a random elite cognitive learning (RECL) strategy was devised by randomly selecting superior particles to direct their evolution with the selection probabilities calculated by a non-linear weight function related to the fitness rankings of superior particles. By contrast, for superior particles, a stochastic dominant cognitive learning (SDCL) strategy was designed based on a stochastic competition mechanism to trigger the update of relatively poor superior particles. With the two heterogeneous learning schemes, inferior particles focus on exploring the search space for seeking optimal regions, while superior particles concentrate on exploiting the optimal zones they locate at for finding accurate solutions. Furthermore, we devised an adaptive swarm partition strategy to dynamically separate particles into the two categories. With this adaptive strategy, HCLPSO gradually narrows down the learning range of inferior particles and the exploitation range of superior particles. As a result, HCLPSO gradually switches from exploring the search space to exploiting the found optimal regions. With the cohesive synergy of the devised two techniques, HCLPSO is expected to traverse the vast space with a proper balance between intensification and diversification and thus is capable of effectively tackling LSOPs.

A series of experiments have been executed on the publicly acknowledged CEC2010 and CEC2013 LSOP benchmark suites by comparing HCLPSO with 8 state-of-the-art approaches for LSOPs. Extensive comparisons have validated that HCLPSO performs considerably competitively with or even far better than the compared large-scale optimizers regarding both the convergence speed and the solution quality. In particular, empirical studies found that HCLPSO is particularly effective in solving complicated LSOPs, like partially separable LSOPs, overlapping LSOPs, and fully non-separable LSOPs. In addition, the good scalability of HCLPSO to solve LSOPs with different dimensionalities has also been demonstrated by experiments conducted on the CEC2010 LSOP suite with the dimension size ranging from 200 to 2000.

In the future, we aim to devise adaptive mechanisms for HCLPSO to relieve its sensitivity to the control factor β and the swarm size NP to further improve its optimization performance, and at the same time, relieve users from tedious parameter fine-tuning so as to promote its usability. Another direction is to employ HCLPSO to tackle practical optimization problems in real-world engineering.

CRediT authorship contribution statement

En Zhang: Conceptualization, Supervision, Methodology, Formal analysis. **Zihao Nie:** Methodology, Software, Validation, Writing – original draft. **Qiang Yang:** Methodology, Supervision, Formal analysis, Writing – original draft. **Yiqiao Wang:** Formal analysis,

Writing – review & editing. **Dong Liu:** Formal analysis, Writing – review & editing. **Sang-Woon Jeon:** Formal analysis, Writing – review & editing. **Jun Zhang:** Conceptualization, Methodology, Writing – review & editing.

Declaration of Competing Interest

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

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62006124, and 62072160, in part by the Natural Science Foundation of Jiangsu Province under Project BK20200811, and in part by the National Research Foundation of Korea under Grant NRF-2021H1D3A2A01082705.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ins.2023.03.086.

References

[1] X. Zhang, Q. Lin, Three-Learning Strategy Particle Swarm Algorithm for Global Optimization Problems, Inf. Sci. 593 (2022) 289–313.
[2] Q. Yang, L. Hua, X. Gao, D. Xu, Z. Lu, S.-W. Jeon, J. Zhang, Stochastic Cognitive Dominance Leading Particle Swarm Optimization for Multimodal Problems, Mathematics 10 (2022) 761.
[3] W.Y. Dong, R.R. Zhang, Stochastic Stability Analysis of Composite Dynamic System for Particle Swarm Optimization, Inf. Sci. 592 (2022) 227–243.
[4] J. Kennedy, R. Eberhart, Particle Swarm Optimization, in: Proceedings of ICNN'95 – International Conference on Neural Networks, vol. 1944, 1995, pp. 1942–1948.
[5] H.-Q. Xu, S. Gu, Y.-C. Fan, X.-S. Li, Y.-F. Zhao, J. Zhao, J.-J. Wang, A Strategy Learning Framework for Particle Swarm Optimization Algorithm, Inf. Sci. 619 (2023) 126–152.
[6] Q. Yang, Y.-W. Bian, X. Gao, D. Xu, Z. Lu, S.-W. Jeon, J.J.M. Zhang, Stochastic Triad Topology Based Particle Swarm Optimization for Global Numerical Optimization, Mathematics (2022).
[7] Q. Yang, X. Guo, X.-D. Gao, D.-D. Xu, Z.-Y. Lu, Differential Elite Learning Particle Swarm Optimization for Global Numerical Optimization, Mathematics 10 (2022) 1261.
[8] Q. Yang, Y. Jing, X. Gao, D. Xu, Z. Lu, S.-W. Jeon, J. Zhang, Predominant Cognitive Learning Particle Swarm Optimization for Global Numerical Optimization, Mathematics 10 (2022) 1620.
[9] Q. Yang, W.-N. Chen, J.D. Deng, Y. Li, T. Gu, J. Zhang, A Level-Based Learning Swarm Optimizer for Large-Scale Optimization, IEEE Trans. Evol. Comput. 22 (2018) 578–594.
[10] J.-R. Jian, Z.-H. Zhan, J. Zhang, Large-Scale Evolutionary Optimization: A Survey and Experimental Comparative Study, Int. J. Mach. Learn. Cybern. 11 (2020) 729–745.
[11] R. Lan, Y. Zhu, H. Lu, Z. Liu, X. Luo, A Two-Phase Learning-Based Swarm Optimizer for Large-Scale Optimization, IEEE Trans. Cybern. 51 (2021) 6284–6293.
[12] Q. Yang, K.-X. Zhang, X.-D. Gao, D.-D. Xu, Z.-Y. Lu, S.-W. Jeon, J. Zhang, A Dimension Group-Based Comprehensive Elite Learning Swarm Optimizer for Large-Scale Optimization, Mathematics 10 (2022) 1072.
[13] S. Zhao, D. Wang, Elite-Ordinary Synergistic Particle Swarm Optimization, Inf. Sci. 609 (2022) 1567–1587.
[14] Z. Lu, S. Liang, Q. Yang, B. Du, Evolving Block-Based Convolutional Neural Network for Hyperspectral Image Classification, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–21.
[15] T. Li, Z.-H. Zhan, J.-C. Xu, Q. Yang, Y.-Y. Ma, A Binary Individual Search Strategy-Based Bi-Objective Evolutionary Algorithm for High-Dimensional Feature Selection, Inf. Sci. 610 (2022) 651–673.
[16] M.N. Omidvar, X. Li, X. Yao, A Review of Population-Based Metaheuristics for Large-Scale Black-Box Global Optimization: Part A, IEEE Trans. Evol. Comput. (2021) 1-1.
[17] C. Huang, X. Zhou, X. Ran, Y. Liu, W. Deng, W. Deng, Co-Evolutionary Competitive Swarm Optimizer with Three-Phase for Large-Scale Complex Optimization Problem, Inf. Sci. 619 (2023) 2–18.
[18] R. Cheng, Y. Jin, A Competitive Swarm Optimizer for Large Scale Optimization, IEEE Trans. Cybern. 45 (2015) 191–204.
[19] Q. Yang, W. Chen, T. Gu, H. Zhang, J.D. Deng, Y. Li, J. Zhang, Segment-Based Predominant Learning Swarm Optimizer for Large-Scale Optimization, IEEE Trans. Cybern. 47 (2017) 2896–2910.
[20] Q. Yang, Y. Zhu, X. Gao, D. Xu, Z. Lu, Elite Directed Particle Swarm Optimization with Historical Information for High-Dimensional Problems, Mathematics 10 (2022) 1384.
[21] F.v.d. Bergh, A.P. Engelbrecht, A Cooperative Approach to Particle Swarm Optimization, IEEE Trans. Evol. Comput. 8 (2004) 225–239.
[22] X. Li, X. Yao, Cooperatively Coevolving Particle Swarms for Large Scale Optimization, IEEE Trans. Evol. Comput. 16 (2012) 210–224.
[23] F. Wang, X. Wang, S. Sun, A Reinforcement Learning Level-Based Particle Swarm Optimization Algorithm for Large-Scale Optimization, Inf. Sci. 602 (2022) 298–312.
[24] Q. Yang, W.N. Chen, T. Gu, H. Jin, W. Mao, J. Zhang, An Adaptive Stochastic Dominant Learning Swarm Optimizer for High-Dimensional Optimization, IEEE Trans. Cybern. 52 (2020) 1960–1976.
[25] R. Cheng, Y. Jin, A Social Learning Particle Swarm Optimization Algorithm for Scalable Optimization, Inf. Sci. 291 (2015) 43–60.
[26] X. Ma, X. Li, Q. Zhang, K. Tang, Z. Liang, W. Xie, Z. Zhu, A Survey on Cooperative Co-Evolutionary Algorithms, IEEE Trans. Evol. Comput. 23 (2019) 421–441.
[27] Z.J. Wang, Z.H. Zhan, S. Kwong, H. Jin, J. Zhang, Adaptive Granularity Learning Distributed Particle Swarm Optimization for Large-Scale Optimization, IEEE Trans. Cybern. 51 (2021) 1175–1188.

[28] Z. Yang, K. Tang, X. Yao, Large Scale Evolutionary Optimization Using Cooperative Coevolution, Inf. Sci. 178 (2008) 2985–2999.
[29] Y. Sun, M. Kirley, S.K. Halgamuge, A Recursive Decomposition Method for Large Scale Continuous Optimization, IEEE Trans. Evol. Comput. 22 (2018) 647–661.
[30] X. Ma, Z. Huang, X. Li, L. Wang, Y. Qi, Z. Zhu, Merged Differential Grouping for Large-scale Global Optimization, IEEE Trans. Evol. Comput. (2022) 1-1.
[31] A. Chen, Z. Ren, W. Guo, Y. Liang, Z. Feng, An Efficient Adaptive Differential Grouping Algorithm for Large-Scale Black-Box Optimization, IEEE Trans. Evol. Comput. (2022) 1-1.
[32] K. Tang, X. Li, P. Suganthan, Z. Yang, T. Weise, 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, & Nanyang Technological University, (2009).
[33] X. Li, K. Tang, M.N. Omidvar, Z. Yang, K. Qin, H. China, Benchmark Functions for the CEC 2013 Special Session and Competition on Large-Scale Global Optimization, Technical Report, Evolutionary Computation and Machine Learning Group, RMIT University, Australia, 7 (2013) 8.
[34] J.R. Jian, Z.G. Chen, Z.H. Zhan, J. Zhang, Region Encoding Helps Evolutionary Computation Evolve Faster: A New Solution Encoding Scheme in Particle Swarm for Large-Scale Optimization, IEEE Trans. Evol. Comput. 25 (2021) 779–793.
[35] H. Deng, L. Peng, H. Zhang, B. Yang, Z. Chen, Ranking-Based Biased Learning Swarm Optimizer for Large-Scale Optimization, Inf. Sci. 493 (2019) 120–137.
[36] M. Sheng, Z. Wang, W. Liu, X. Wang, S. Chen, X. Liu, A Particle Swarm Optimizer with Multi-Level Population Sampling and Dynamic P-Learning Mechanisms for Large-Scale Optimization, Knowl.-Based Syst. 242 (2022), 108382.
[37] D. Li, W. Guo, A. Lerch, Y. Li, L. Wang, Q. Wu, An Adaptive Particle Swarm Optimizer with Decoupled Exploration and Exploitation for Large Scale Optimization, Swarm Evol. Comput. 60 (2021), 100789.
[38] H. Wang, M. Liang, C. Sun, G. Zhang, L. Xie, Multiple-Strategy Learning Particle Swarm Optimization for Large-Scale Optimization Problems, Complex & Intelligent Systems 7 (2021) 1–16.
[39] M.A. Potter, The Design and Analysis of A Computational Model of Cooperative Coevolution, George Mason University, 1997.
[40] T. Ray, X. Yao, A Cooperative Coevolutionary Algorithm with Correlation Based Adaptive Variable Partitioning, in: 2009 IEEE Congress on Evolutionary Computation, 2009, pp. 983-989.
[41] Y. Mei, M.N. Omidvar, X. Li, X. Yao, A Competitive Divide-and-Conquer Algorithm for Unconstrained Large-Scale Black-Box Optimization, ACM Trans. Math. Softw. 42 (2016). Article 13.
[42] M.N. Omidvar, M. Yang, Y. Mei, X. Li, X. Yao, DG2: A Faster and More Accurate Differential Grouping for Large-Scale Black-Box Optimization, IEEE Trans. Evol. Comput. 21 (2017) 929–942.
[43] M. Yang, A. Zhou, C. Li, X. Yao, An Efficient Recursive Differential Grouping for Large-Scale Continuous Problems, IEEE Trans. Evol. Comput. 25 (2021) 159–171.
[44] H. Liu, Y. Wang, N. Fan, A Hybrid Deep Grouping Algorithm for Large Scale Global Optimization, IEEE Trans. Evol. Comput. 24 (2020) 1112–1124.
[45] X.Y. Zhang, Y.J. Gong, Y. Lin, J. Zhang, S. Kwong, J. Zhang, Dynamic Cooperative Coevolution for Large Scale Optimization, IEEE Trans. Evol. Comput. 23 (2019) 935–948.
[46] Z.J. Wang, Z.H. Zhan, W.J. Yu, Y. Lin, J. Zhang, T.L. Gu, J. Zhang, Dynamic Group Learning Distributed Particle Swarm Optimization for Large-Scale Optimization and Its Application in Cloud Workflow Scheduling, IEEE Trans. Cybern. 50 (2020) 2715–2729.
[47] Q. Yang, W.N. Chen, T. Gu, H. Zhang, H. Yuan, S. Kwong, J. Zhang, A Distributed Swarm Optimizer With Adaptive Communication for Large-Scale Optimization, IEEE Trans. Cybern. 50 (2020) 3393–3408.
[48] Y. Sun, M.N. Omidvar, M. Kirley, X. Li, Adaptive Threshold Parameter Estimation with Recursive Differential Grouping for Problem Decomposition, in: Proceedings of the Genetic and Evolutionary Computation Conference, Association for Computing Machinery, Kyoto, Japan, 2018, pp. 889–896.
[49] Q. Yang, J.-Q. Yan, X.-D. Gao, D.-D. Xu, Z.-Y. Lu, J. Zhang, Random Neighbor Elite Guided Differential Evolution for Global Numerical Optimization, Inf. Sci. 607 (2022) 1408–1438.
[50] J. Zhang, D. Chen, Q. Yang, Y. Wang, D. Liu, S.-W. Jeon, J. Zhang, Proximity Ranking-Based Multimodal Differential Evolution, Swarm Evol. Comput. 78 (2023), 101277.