Complex & Intelligent Systems (2023) 9:5467–5500
https://doi.org/10.1007/s40747-023-00993-w

ORIGINAL ARTICLE

A random elite ensemble learning swarm optimizer for high-dimensional optimization

Qiang Yang[1] · Gong-Wei Song[1] · Xu-Dong Gao[1] · Zhen-Yu Lu[1] · Sang-Woon Jeon[2] · Jun Zhang[2]

Received: 18 October 2022 / Accepted: 27 January 2023 / Published online: 23 March 2023

Abstract
High-dimensional optimization problems are increasingly pervasive in real-world applications nowadays and become harder and harder to optimize due to increasingly interacting variables. To tackle such problems effectively, this paper designs a random elite ensemble learning swarm optimizer (REELSO) by taking inspiration from human observational learning theory. First, this optimizer partitions particles in the current swarm into two exclusive groups: the elite group consisting of the top best particles and the non-elite group containing the rest based on their fitness values. Next, it employs particles in the elite group to build random elite neighbors for each particle in the non-elite group to form a positive learning environment for the non-elite particle to observe. Subsequently, the non-elite particle is updated by cognitively learning from the best elite among the neighbors and collectively learning from all elites in the environment. For one thing, each non-elite particle is directed by superior ones, and thus the convergence of the swarm could be guaranteed. For another, the elite learning environment is randomly formed for each non-elite particle, and hence high swarm diversity could be maintained. Finally, this paper further devises a dynamic partition strategy to divide the swarm into the two groups dynamically during the evolution, so that the swarm gradually changes from exploring the immense solution space to exploiting the found optimal areas without serious diversity loss. With the above mechanisms, the devised REELSO is expected to explore the search space and exploit the found optimal areas properly. Abundant experiments on two popularly used high-dimensional benchmark sets prove that the devised optimizer performs competitively with or even significantly outperforms several state-of-the-art approaches designed for high-dimensional optimization.

Keywords Particle swarm optimization · Large-scale optimization · Random elite ensemble learning swarm optimizer · Ensemble learning · Cognitive learning · High-dimensional problems

Introduction

High-dimensional optimization problems, involving hundreds or thousands of variables, are exceedingly common in various fields, such as complex networks optimization [1, 2], order scheduling [3], industrial copper burdening system optimization [4], power control optimization [5, 6], joint deployment and task scheduling optimization [7], neural network optimization [8], and community detection [9]. Compared with low-dimensional problems, the complexity of high-dimensional problems increases exponentially, leading to that they are considerably difficult to optimize [10–13]. To be specific, for one thing, the solution space of high-dimensional problems increases exponentially as the dimensionality increases [14]. Consequently, it is very challenging to seek the globally optimal solution efficiently in such vast space. For another, it also often occurs that a variety of spacious and flat local regions or saddle areas exist in the high-dimensional environment [10, 15, 16]. As a consequence, it is considerably impeditive to locating the global optimum of a high-dimensional problem due to the strong and greedy attraction of these local basins or saddle regions.

The above challenges of high-dimensional optimization seriously degrade the effectiveness and efficiency of traditional optimization methods designed for low-dimensional problems [11, 17–19]. Such a phenomenon is often called

✉ Qiang Yang
mmmyq@126.com

1 School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing 210044, China

2 Department of Electrical and Electronic Engineering, Hanyang University, Ansan 15588, Korea

"the curse of dimensionality" [10]. Since high-dimensional problems are ubiquitous in various fields, lacking effective optimization methods to deal with them hinders the development of related fields and industries. Consequently, there is an increasing demand for developing effective and efficient large-scale optimization methods, which has drawn plenty of attention from researchers.

As a typical gradient-free heuristic method, particle swarm optimization (PSO) [20, 21] has exhibited great success in solving optimization problems with different properties, such as multimodal optimization problems [22–24], constrained optimization problems [25–29], multi-objective optimization problems [30], and expensive optimization problems [31]. Besides, it has been broadly employed to cope with real-world optimization problems [32–36]. Nevertheless, most existing PSO algorithms are mainly devised to deal with low-dimensional optimization problems [11, 37]. In face of high-dimensional optimization problems, most existing PSOs face the dilemma of either falling into local areas or not being able to find high-quality solutions under the afforded number of fitness evaluations [38–40].

To get out of these predicaments, researchers have devoted themselves to devising simple yet effective PSOs for high-dimensional optimization, and thus, many remarkable large-scale PSOs have emerged [12, 41, 42]. As far as we are concerned, existing research on large-scale PSO roughly proceeds in two main directions [43]: (1) decomposition-based PSOs [44–46], which are also named cooperative co-evolutionary PSOs (CCPSOs), and (2) non-decomposition-based PSOs [47–49].

The first kind of large-scale PSOs are based on the thought of the divide-and-conquer approach [41]. Specifically, CCPSOs [45] first utilize variable decomposition strategies to partition a large-scale problem into a number of small-scale sub-problems. Next, these sub-problems are separately optimized by PSO. At last, the optimal values of the variables in all sub-problems are integrated to construct the optimal solution to the high-dimensional problem. In this manner, existing PSOs developed for small-scale optimization problems can be utilized to deal with large-scale optimization [45, 46]. However, during the optimization, interacting variables usually interfere with each other [50]. Therefore, ideally, interacting variables should be placed into the same sub-problem to optimize. Nevertheless, without prior knowledge of correlations between variables, it is very challenging to partition variables into sub-problems accurately. To this end, researchers have paid extensive attention to designing novel decomposition methods to attempt to accurately separate a large-scale problem into exclusive small-scale sub-problems [51, 52].

Although CCPSOs have been testified to be effective in coping with high-dimensional optimization problems to a certain extent, their optimization performance heavily depends on the variable decoupling strategy. Once correlated variables are partitioned into different sub-problems, their effectiveness deteriorates sharply [41]. Given this, the robustness of CCPSOs is usually limited, and thus some researchers direct their attention to another direction by devising non-decomposition-based PSOs [47].

Unlike CCPSOs, all variables are still simultaneously optimized together like traditional PSOs in non-decomposition-based PSOs [53, 54]. The key to effectuating non-decomposition-based PSOs in coping with high-dimensional optimization problems is to design simple yet effective learning mechanisms for particles, so that they could search the exponentially increased solution space appropriately [55, 56]. In this direction, taking inspiration from intelligent collective behaviors of human beings and natural animals, researchers have devised a lot of new learning strategies, like the competition learning strategy [47], the level-based learning scheme [49] and the phased learning mechanism [55]. Among these effective strategies, it is found that instead of using historical information (like the personal best experience or the global best experience) to guide the update of particles, these methods directly utilize superior individuals in the current swarm to direct the update of inferior ones along with the superior individuals being not updated. In this way, the guiding exemplars for different particles are distinct in the same iteration and they are likely distinct for the same particle in different iterations as well. Therefore, high swarm diversity could be maintained during the evolution, which is profitable for particles to search the solution space dispersedly and avoid falling into local areas [57].

Though existing large-scale PSOs have exhibited promising performance to a large extent in tackling certain kinds of high-dimensional optimization problems, they still encounter limitations in coping with complex large-scale problems, particularly those with complicated interacting variables [10, 14]. Hence, how to effectively further promote the capability of PSO in settling complicated large-scale problems still remains a challenging and open issue and deserves in-depth research.

In the human observational learning theory proposed by Bandura [58], the behavior and cognition of an individual are usually influenced by his surroundings [59]. By taking the observational and causal learning schemes, human beings usually watch the actions and talks of their surroundings, gain information to discover how things work, and learn to do things by themselves [60]. Taking inspiration from such learning theory, this paper proposes a random elite ensemble learning swarm optimizer (REELSO) to solve high-dimensional optimization problems. Specifically, to provide positive environment for particles to learn, this paper first separates particles in the swarm into two exclusive groups, namely the elite group constituted by the top ranked individuals in terms of fitness, and the non-elite group containing the rest ones. Then, for each particle in the non-elite group, several elites are stochastically selected from the elite group to form a random elite neighbor region (namely the learning environment). By this means, each individual in the non-elite group is surrounded by elites and thus can acquire positive learning. Subsequently, by watching the behaviors of these elites, the non-elite particle is guided to search the solution space by cognitively imitating the best elite and collectively learning from all elites in the neighbor region.

To sum up, the main novelty of this paper is summarized as follows:

(1) A random elite ensemble learning strategy (REEL) is devised to direct the evolution of particles, which is inspired by the human observational learning theory. As mentioned above, each non-elite particle is surrounded by elite individuals randomly chosen from the elite group. Such good environment makes positive influence on the non-elite particle and thus is beneficial for improving its learning ability. On the one hand, by watching the promising behaviors of these elites, the non-elite particle could fly through the vast solution space fast to find promising zones. On the other hand, the elites used to form the learning environment of each non-elite particle are randomly selected from the elite group. As a result, for different non-elite particles, the elite environment is different and thus the observational learning is also different. This indicates that the guiding exemplars to direct the evolution of different non-elite individuals are likely different in the same generation. Besides, the guiding exemplars are also distinct for the same non-elite individual in different iterations. Such randomness in the construction of the neighbor elite environment provides high diversity for the swarm.

(2) A dynamic partition strategy is further devised to divide the swarm into the two exclusive groups dynamically during the evolution. Since particles in the elite group are utilized to construct the neighbor elite environment, the elite group size makes significant influence on the performance of the optimizer, leading to this optimizer being sensitive to the elite group size. To resolve this dilemma, this paper further develops a dynamic partition strategy by gradually reducing the elite group size to dynamically separate the swarm into the two groups as the evolution iterates. In this way, more non-elite individuals are evolved by gradually fewer elite ones, leading to that the non-elite individuals gradually concentrate on intensive learning. As a result, the optimizer gradually changes from exploring the solution space to exploiting the found optimal zones without seriously sacrificing swarm diversity as the iteration goes on.

(3) With the above two mechanisms, the designed REELSO is expected to compromise search intensification and diversification of the swarm at both the particle level (by the REEL strategy) and the swarm level (by the dynamic partition strategy). By the above means, REELSO hopefully explores the vast solution space and exploits the found optimal areas appropriately and thus expectedly obtains good performance in solving high-dimensional optimization problems.

To verify the effectiveness of REELSO, comprehensive experiments are extensively performed on the public and popular CEC'2010 [40] and CEC'2013 [61] high-dimensional problem sets by virtue of comparing REELSO with totally 14 popular and state-of-the-art evolutionary optimizers for high-dimensional optimization. In addition, to investigate the capability of REELSO to cope with optimization problems with higher dimensionality, comparative experiments are further executed on the CEC'2010 high-dimensional problem set with 2000 dimensions.

The remaining part of this paper is arranged as follows. In the following section, the classical PSO and recent large-scale PSOs are reviewed. Next, the detailed elucidation of the proposed REELSO is presented. Subsequently, comparative experiments are executed to verify the effectiveness of REELSO. Finally, the conclusion of this paper is provided.

Related works on large-scale PSO

Without loss of generality, minimization problems containing D variables are considered in this paper. Furthermore, the objective value evaluated on the minimization problem is taken as the fitness of a particle.

Classical PSO

The update formula of each particle in the classical PSO is listed below:

$$\boldsymbol{v}_i = w\boldsymbol{v}_i + c_1 r_1(\boldsymbol{x}_{i\_\text{pbest}} - \boldsymbol{x}_i) + c_2 r_2(\boldsymbol{x}_{\text{gbest}} - \boldsymbol{x}_i), \tag{1}$$

$$\boldsymbol{x}_i = \boldsymbol{x}_i + \boldsymbol{v}_i, \tag{2}$$

where $\boldsymbol{v}_i$ and $\boldsymbol{x}_i$ denote the velocity vector and the position vector of the ith particle, respectively. w is termed as the inertia weight. $\boldsymbol{x}_{i\_\text{pbest}}$ represents the best solution located by the ith particle so far, while $\boldsymbol{x}_{\text{gbest}}$ is the best solution located by the entire swarm so far, which is actually the best one among all best positions found by all particles. c_1 and c_2 are two acceleration coefficients, while r_1 and r_2 are two real vectors with each element randomly and uniformly generated within [0, 1].

In the literature [20], ($\boldsymbol{x}_{i\_\text{pbest}} - \boldsymbol{x}_i$) in Eq. (1) is often considered as the individual cognitive learning behavior, which is usually a way for the updated particle to learn from its own successful experience. ($\boldsymbol{x}_{\text{gbest}} - \boldsymbol{x}_i$) in Eq. (1) is often referred to as the swarm social learning behavior, which is usually a process where the updated particle learns from the successful experience of the swarm.

The classical learning strategy shown in Eq. (1) utilizes the historically successful evolutionary information to direct the update of particles. Though it has witnessed great success in solving simple optimization problems [21, 62], like unimodal problems, its effectiveness and efficiency drastically degrades when solving multimodal problems [22, 63, 64]. This is mainly because of the greedy dragging of the found best position ($\boldsymbol{x}_{\text{gbest}}$) by the swarm. Aiming at improving the optimization performance of PSO in coping with multimodal problems, a lot of researchers have poured extensive attention to designing novel effective learning strategies by taking inspiration from intelligent behaviors of gregarious animals in nature and human society. As a result, many remarkable learning strategies have sprung up [65], such as predominant cognitive learning [66], comprehensive learning [67], orthogonal learning [19, 68], scatter learning [64], differential elite learning [69], and rank-based learning [70].

With the advance of PSO, it is gradually realized that it is the learning strategy for directing the update of particles that is the most crucial part of PSO [71]. However, most learning strategies in existing PSO variants are mainly designed for solving small-scale optimization problems. As the dimensionality increases, most existing PSO variants lose their effectiveness or even feasibility [72] on account of the "curse of dimensionality". As a consequence, to tackle the aforementioned challenges of large-scale optimization with high effectiveness and efficiency, researchers concentrate on developing novel learning strategies suitable for large-scale optimization problems. Broadly speaking, research on PSO variants for high-dimensional optimization is mainly categorized into two classes, namely decomposition-based large-scale PSOs [45, 46, 55, 56] and non-decomposition-based large-scale PSOs [47, 50, 51, 73], which are elucidated in the next two subsections, respectively.

Decomposition-based large-scale PSO

Decomposition-based PSOs are also named cooperative co-evolutionary PSOs (CCPSOs). The basic thought of CCPSOs is to utilize the divide-and-conquer technique to decompose a high-dimensional problem into several exclusive low-dimensional sub-problems and then optimize each sub-problem separately by employing low-dimensional PSOs [41]. In this way, traditional PSOs designed for low-dimensional problems can be utilized to solve high-dimensional optimization problems.

Bergh and Engelbrecht took the first attempt to design CCPSO [45], where two cooperative PSO models, namely CCPSO-S_K and CCPSO-H_K, were proposed. The former first uses a random variable decomposition method to separate D variables into K variable groups with each consisting of D/K variables. Then, it adopts traditional PSO to seek the optimal values of each variable group. The latter model alternatively utilizes traditional PSO to optimize all variables together or adopts CCPSO-S_K to separately optimize decomposed groups of variables during the evolution. Though CCPSO has shown effectiveness on some high-dimensional problems [39], its performance heavily relies on the setting of the number of groups (namely K). To circumvent this predicament, an improved version of CCPSO, named CCPSO2, was proposed in Ref. [46] by predefining a set of group numbers. Then, during the evolution, CCPSO2 first randomly chooses a group number from the set at each generation and then divides variables randomly into groups based on the selected group number. Both CCPSO and CCPSO2 adopt the random decomposition scheme to partition variables into groups. Nevertheless, this strategy does not explicitly consider the correlations between variables and thus both CCPSO and CCPSO2 show poor performance on non-separable problems with many interacting variables.

In general, the optimization of interacting variables often interferes with each other. Therefore, ideally, interacting variables should be put into the same group to optimize. This indicates that variable decomposition plays a key role in CCPSOs [50]. As a result, in recent years, researches on decomposition-based evolutionary algorithms, including CCPSOs, mainly focus on excogitating effective decomposition strategies to divide a large-scale problem into small-scale sub-problems as accurately as possible by discovering the correlations between variables and thus a lot of remarkable decomposition strategies have been developed [41, 74, 75].

As for the research on decomposition methods, the most typical one is the differential grouping (DG) strategy [50]. Specifically, this method separates variables into several groups by detecting pairwise correlations based on the partial difference in function values of the associated shifted solutions. Nevertheless, DG can only identify direct interactions between variables with indirect interactions ignored. As a result, its performance is limited on problems containing indirectly correlated variables. To alleviate this dilemma, an extended DG (XDG) was devised in Ref. [74] by discovering both indirect and direct relationships among variables. Further, to resolve the issue that DG (including XDG) is sensitive to its parameters, a global DG (GDG) was put forward in Ref. [75] by devising an adaptive parameter adjustment strategy. Another shortcoming of DG and its variants is that they take too many fitness evaluations (up to $O(D^2)$ fitness evaluations) to discover the interactions between variables. This results in that given limited fitness evaluations, the optimization of sub-problems is not sufficient enough.

To decline the fitness evaluation consumption in the decomposition stage, a fast DG, named DG2 [76], was proposed by reusing the sampled points to detect the interactions between variables. In particular, it saves half of fitness evaluations on fully separable problems. Enlightened by the mechanism of binary search, Sun et al. devised a recursive DG (RDG) [51], which finally takes $O(D\log(D))$ fitness evaluations in the decomposition stage. Subsequently, Sun et al. further devised an adaptive threshold estimation method for RDG, leading to RDG2 [77]. Based on the analysis of the binary search process and the variable interaction detection in RDG, Yang et al. proposed an efficient RDG [78] by fully utilizing the historical information to detect correlations between variables. This strategy avoids some redundant variable interaction detection and thus reduces the consumption of fitness evaluations to a large extent.

Although the aforementioned decomposition strategies could help CCPSOs achieve promising performance in dealing with high-dimensional problems, they still encounter limitations in handling complex high-dimensional optimization problems. First, on the basis of the theorem of No Free Lunch, a universal decomposition method does not exist to accurately decompose variables for all kinds of high-dimensional optimization problems. Second, for large-scale problems containing overlapping interactions between variables, most existing decomposition methods would place all mutually interacting variables into one same group. As a consequence, there might be a large group containing a lot of interacting variables. Extremely, the worst case is that all variables are placed into only one group. In this situation, on the one hand, it is difficult for CCPSOs to optimize the decomposed sub-problems effectively; on the other hand, a lot of fitness evaluations are wasted in the decomposition stage.

Non-decomposition-based large-scale PSO

To alleviate the above limitation of CCPSOs, researchers turn to finding breakthrough of PSO in solving high-dimensional optimization problems in another direction, namely non-decomposition-based PSOs. Distinguished from CCPSOs, this kind of large-scale PSO variants still optimize all variables simultaneously like the canonical PSO. To conquer “the curse of dimensionality”, the key to non-decomposition-based PSOs is to excogitate novel learning mechanisms with high efficacy to update particles, such that they could search the immense solution space properly during the evolution. To this end, taking inspiration from the collective behaviors of human beings and natural animals, researchers have designed various effective learning schemes for PSO to tackle high-dimensional problems [55, 79, 80]. Since a lot of non-decomposition-based large-scale PSOs have been proposed in the literature, it is hardly possible to review them all. Therefore, this subsection only reviews some representative and latest methods.

In the early research, based on the observation that multiple populations could afford high diversity for species to evolve, a dynamic multi-swarm PSO was devised in Ref. [80] by dynamically separating the swarm into a number of smaller sub-swarms in each generation and then evolving each sub-swarm separately but collaboratively to search the vast solution space. Subsequently, Cheng et al. devised a multi-swarm PSO according to a feedback mechanism to strengthen the optimization capability of PSO [81]. In Ref. [82], a hybrid PSO was devised by combining PSO with the crossover operator and the mutation operator in genetic algorithms (GA), such that a good balance between exploration and exploitation can be maintained.

Inspired by the competitive behavior in human society, a competitive swarm optimizer (CSO) was proposed in [47]. In this method, particles are first randomly arranged into pairs. Subsequently, in each pair of particles, this method does not update the winner, but updates the loser by using the winner and the mean position of the swarm. Getting hints from the social behavior of animals, Cheng et al. developed a social learning PSO (SL-PSO) [48]. Specifically, this optimizer first assigns a learning probability to each particle, which is calculated on the basis of its fitness ranking. Subsequently, each particle is updated probabilistically by using a random predominant particle and the center of the swarm. Particularly, different from traditional PSOs, which adopt historical evolutionary information (like personal best positions ***pbest***s, global best position ***gbest*** or neighbor best positions ***nbest***s) to direct the evolution of particles, both CSO and SL-PSO utilize superior individuals to direct the evolution of inferior ones. Because particles are usually updated generation by generation, both CSO and SL-PSO could maintain high swarm diversity during the evolution and hence they achieve good performance in handling high-dimensional optimization problems.

Subsequently, inspired by the comprehensive learning scheme devised for small-scale optimization problems [67], a segment-based predominant learning mechanism for PSO was devised, resulting in a segment-based predominant learning swarm optimizer (SPLSO) [83]. This optimizer first randomly divides the dimensions into several exclusive segments for each inferior particle and then randomly chooses a superior particle to update every dimension segment of the inferior particle. In this way, several different predominant particles can be employed to guide the update of each inferior one. Inspired by the teaching theory in pedagogy, the authors in [49] proposed a level-based learning scheme for PSO, resulting in a level-based learning swarm optimizer (LLSO) [49]. This algorithm first partitions particles into several levels according to their fitness. Next, particles in lower levels are evolved by two different superior ones randomly selected from two different higher levels. By selecting guiding exemplars with maximized fitness difference to each updated particle, a ranking-based biased PSO was proposed in Ref. [84] by devising two kinds of learning schemes, that is the ranking paired learning scheme and the biased center learning scheme. The former learning mechanism updates worse particles by employing better ones to afford fast convergence, while the latter learning method updates each particle by utilizing a weighted center of the whole swarm to enhance the swarm diversity.

Recently, taking hints from the collaborative behaviors of human beings, Lan et al. proposed a two-phase learning technique for PSO, resulting in a two-phase learning swarm optimizer (TPLSO) [55]. Specifically, the learning of particles is separated into two phases. In the first phase, particles are randomly combined into triads and the competitive strategy is adopted to evolve the members of each triad. In the second phase, several top best particles are selected from the current swarm and then they are evolved by learning from each other to exploit the found optimal regions. Further, a stochastic dominant learning strategy was proposed in [56], leading to a stochastic dominant learning swarm optimizer (SDLSO). In this algorithm, for each particle to be updated, two distinct particles are first randomly chosen from the swarm, and then, only when the two selected particles are superior to this particle, it is updated by learning from the two selected superiors; otherwise, it enters directly the next generation. Besides, to well balance exploration and exploitation during evolution, a learning structure aiming at decoupling intensification and diversification was developed in [57] for PSO to deal with high-dimensional optimization. To be specific, the authors first designed a diversification learning scheme to guide particles to sparse regions according to a measurement used to evaluate local sparseness degree, and then devised an adaptive intensification learning mechanism to update particles by adjusting the fitness differences between exemplars.

Additionally, in recent years, researchers have also paid attention to developing distributed learning strategies for PSO by incorporating distributed computing techniques. For instance, an adaptive granularity learning distributed PSO was devised in Ref. [85]. In this optimizer, the swarm is first divided into several smaller sub-swarms. Then, the master–slave distributed model is adopted to evolve the sub-swarms in parallel. In Ref. [54], a distributed elite guided learning swarm optimizer (DEGLSO) was developed by utilizing the master–slave distributed model to evolve multiple small swarms based on an elite guided learning strategy and devising an adaptive communication scheme to exchange evolutionary information among these swarms adaptively.

Although the above mentioned large-scale PSO variants have exhibited promising performance in tackling certain kinds of high-dimensional problems, they still encounter great challenges on complicated large-scale optimization problems [38, 78], like those containing a lot of interacting variables and the ones with numerous local or saddle regions. As a consequence, there is an increasingly urgent demand for simple yet effective large-scale PSO methods in tackling complex high-dimensional optimization problems. This is why the research on PSO for high-dimensional optimization is still a vibrant and ad hoc topic in the computational intelligence community.

To the above end, this paper proposes a random elite ensemble learning swarm optimizer (REELSO) to cope with high-dimensional optimization by taking inspiration from the human observational learning theory proposed by Bandura [58, 60].

Random elite ensemble learning swarm optimizer

To tackle high-dimensional optimization with high effectiveness and efficiency, we seek inspiration from the learning behaviors of human beings. Specifically, according to the learning theory proposed by Bandura et al. [58–60], human beings usually adopt observational learning to imitate the actions and behaviors of others in their surroundings. Then, valuable information could be gained from the surroundings to discover the ways things work, which is in turn employed to direct us to do them by ourselves. This demonstrates that the behaviors of human beings are deeply influenced by our surrounding environment. Taking inspiration from this, this paper proposes a random elite ensemble learning swarm optimizer (REELSO) to tackle high-dimensional optimization, which is elucidated in detail in the following.

Random elite ensemble learning

Considering each particle in the swarm as an individual, this paper devises a random elite ensemble learning (REEL) strategy to let each particle learn effectively. Specifically, given that the swarm maintains NP particles to iteratively search the solution space, the overall framework of the devised REEL mechanism is presented in Fig. 1 and it works as follows:

(1) In each generation, as shown in Fig. 1, to provide positive learning environment for particles, this scheme first divides particles into two separate groups, namely the elite group (represented as $\boldsymbol{EG}$), and the non-elite group (represented as $\boldsymbol{NEG}$). To be specific, given the size of

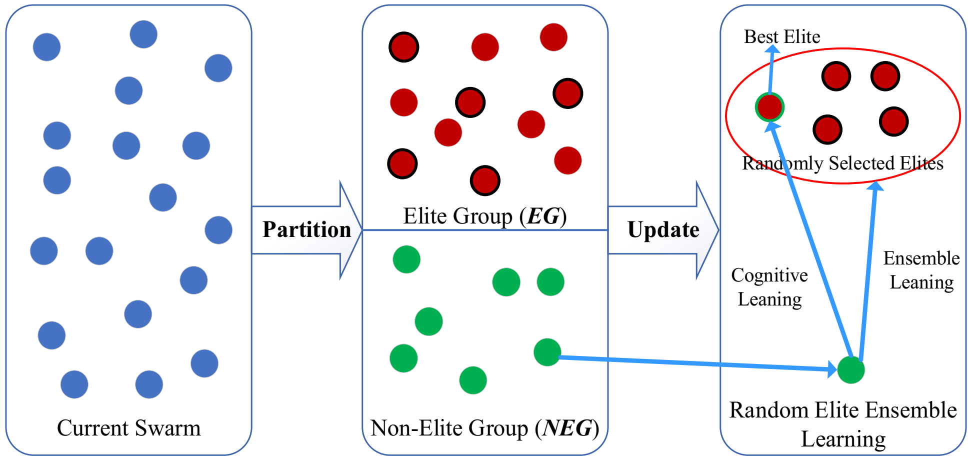


Fig. 1 The framework of REELSO

EG is EGS, ***EG*** is made up by the best EGS particles in the current swarm based on their fitness, while ***NEG*** consists of the remaining (NP-EGS) non-elite particles.

(2) Since the elite particles in ***EG*** are the fittest ones in the current swarm, they preserve valuable evolutionary information to evolve the swarm. Therefore, on the one hand, these elites can be employed to guide the evolution of the particles in ***NEG***, so that each particle in ***NEG*** could take positive observational learning to approach promising areas fast; on the other hand, these elite particles in ***EG*** should not be updated, such that valuable evolutionary information could be preserved to keep the swarm from being trapped into local areas.

(3) To provide positive learning environment for each particle in ***NEG***, as displayed in Fig. 1, the proposed REEL randomly selects a number of different elites from ***EG*** to form a random elite neighbor region for the non-elite particle. In particular, the number of the selected elites is called the elite neighbor size, and denoted as ENS. By this means, each particle in ***NEG*** is surrounded by elites and thus it can acquire positive learning.

(4) Based on the observational learning theory by Bandura [58, 60], each particle in ***NEG*** evolves by watching the behaviors of the elites in its surrounding. In particular, as shown in Fig. 1, REEL updates the non-elite particle by letting it cognitively learn from the best elite and collectively learn from all elites in the neighbor region. To be specific, each non-elite particle $\boldsymbol{x}_i$ is updated as follows:

$$\boldsymbol{v}_i = R_1 \boldsymbol{v}_i + R_2(\boldsymbol{x}_{E_{\text{best}}} - \boldsymbol{x}_i) + R_3 \phi \sum_{t=1}^{\text{ENS}} (\boldsymbol{x}_{E_t} - \boldsymbol{x}_i), \tag{3}$$

$$\boldsymbol{x}_i = \boldsymbol{x}_i + \boldsymbol{v}_i, \tag{4}$$

where $\boldsymbol{v}_i$ and $\boldsymbol{x}_i$ are the velocity vector and the position vector of the ith individual in ***NEG***, respectively. $\boldsymbol{x}_{E\text{best}}$ is the best elite in the random elite neighbor region and is the guiding exemplar in the cognitive learning. $\boldsymbol{x}_{Et}(t = 1, 2, \ldots, \text{ENS})$ is a randomly selected elite from ***EG*** to form the random elite neighbor region (namely, the learning environment of $\boldsymbol{x}_i$) and they are the collective exemplars in the ensemble learning; ENS is the number of the selected elites; R_1, R_2 and R_3 are three real random numbers uniformly generated from [0, 1] and ϕ is a real parameter in [0, 1] used to control the effect of the ensemble learning part on the updated particles.

From Fig. 1 and the above four steps, the features of REEL are summarized as follows:

(1) REEL directly employs the elite particles (namely, the members in ***EG***) in the swarm to guide the update of the non-elite ones (namely, the members in ***NEG***). Therefore, the working principle of REEL is very different from traditional PSOs devised for small-scale problems, which utilize historical evolutionary information (like ***pbest*** and ***gbest***) to direct the update of particles. Thanks to the continuous update of particles generation by generation, the members in both ***EG*** and ***NEG*** are also likely updated generation by generation. As a result, not only the elites used as candidate exemplars to direct the evolution of non-elite particles in ***NEG*** are different in different generations, but also the updated particles are different in different generations.

(2) Each particle in ***NEG*** is surrounded by a number (ENS) of randomly selected elites from ***EG***. On the one hand, based on the learning theory proposed by Bandura [58–60], these elites form a positive learning environment for the associated non-elite particle to observe and imitate. Hence, the non-elite particle could acquire positive learning to approach promising areas. In this way, fast convergence of the swarm to promising areas could be implicitly guaranteed. On the other hand, the elite

neighbor region (namely the learning environment) is constructed by randomly selecting elites from ***EG*** for each non-elite particle. This indicates that for different non-elite particles, their elite neighbor regions are likely different. This matches the human observational learning theory proposed by Bandura [58–60] that individuals could learn different skills and behaviors in different environments. Besides, it also matches the expectation that the non-elite particles should be guided to promising zones fast without sacrificing swarm diversity.

(3) As shown in Eq. (3), REEL utilizes the best elite in the elite neighbor region to direct the cognitive learning and adopts all elite neighbors to direct the ensemble learning of each particle in ***NEG***. On the one hand, the best elite preserves the strongest attraction in the elite neighbor region, and thus by watching its behaviors, the non-elite particle can gain specific skills and approach promising areas fast; on the other hand, the elites in the neighbor region preserve different skills and capabilities, and thus by watching their behaviors, the non-elite particle can gain comprehensive skills to improve its search ability. In particular, the cognitive learning part is mainly responsible for the convergence, because the guiding exemplar in this part is the best elite in the elite neighbor region. On the contrary, the ensemble learning takes charge of the diversity maintenance, because all elites in the region are used to guide the evolution of the associated particle. By this means, it can prevent the updated particle from greedily approaching the areas where the best elite lies. With the collaboration between the cognitive learning and the ensemble learning, the swarm could find promising zones without drastic loss of swarm diversity.

(4) With the above mechanisms, it is found that the proposed REEL could compromise convergence and diversity of the swarm well to explore the vast space and exploit the found promising zones appropriately. Therefore, it is expected that the proposed REEL strategy could help PSO to effectively solve high-dimensional problems.

Adaptive swarm partition

In the proposed REEL strategy, since the elites in ***EG*** are utilized to form the random elite neighbor regions (namely, the learning environment) of the particles in ***NEG***, the size of ***EG***, namely EGS, has great influence on the construction of their learning environment. Specifically, a large EGS exerts the following two influences: (1) The elite group ***EG*** contains a large number of elite particles, and thus the diversity of these elites is high. This leads to that the diversity of the random elite neighbor regions (the learning environments) for non-elite particles is high, and therefore different non-elite particles could learn to seek promising areas diversely. This is quite profitable for particles to explore the large-scale space. (2) The number of non-elite particles in ***NEG*** is small, which indicates that fewer particles are updated during the evolution. In this situation, slow convergence is obtained. On the contrary, a small EGS results in two inverse effects: (1) the number of candidate elites in ***EG*** to form the learning environment of non-elite particles is small. In this situation, the diversity of the random elite neighbor regions is low, leading to that the updated non-elite particles tend to assemble together to exploit the search space. This is profitable for the swarm to exploit promising areas to acquire high-quality solutions. (2) More non-elite particles in ***NEG*** are updated with lower diversity of learning environments. In this case, the swarm may converge fast to promising zones.

In general, in the early evolution period of an EA, high population diversity is usually preferred to fully explore the high-dimensional space to seek promising regions, while in the late evolution stage, good exploitation is usually preferred to intensively exploit the found optimal regions to find high-quality solutions [62, 86]. Based on the above considerations, this paper devises an adaptive EGS adjustment strategy to dynamically partition the swarm into the two groups. Specifically, in each generation, EGS is calculated as follows:

$$\mathrm{EGS}_i = \left[\mathrm{EGS}_{\max} - (\mathrm{EGS}_{\max} - \mathrm{EGS}_{\min}) \times \left(\frac{\mathrm{FEs}_i}{\mathrm{FEs}_{\max}}\right)^{\alpha}\right] \times \mathrm{NP}, \tag{5}$$

where EGS_i is the elite group size in the ith iteration, $\mathrm{EGS}_{\max}$ and $\mathrm{EGS}_{\min}$ are the maximum and the minimum values of EGS. FEs_i is the accumulated number of fitness evaluations consumed before the ith iteration, $\mathrm{FEs}_{\max}$ is the preset maximum number of fitness evaluations, NP is the swarm size, and α is a parameter controlling the decreasing speed of EGS as the evolution goes on. In this paper, we set $\mathrm{EGS}_{\max} = 0.8$*NP, $\mathrm{EGS}_{\max} = 0.4$*NP, and $\alpha = 0.8$ based on investigation experiments conducted in the following section.

From Eq. (5), it is found that as the evolution goes on, the elite group ***EG*** becomes smaller and smaller. This indicates that as the iteration continues, the swarm gradually changes from exploring the high-dimensional space to exploiting the found optimal areas. Specifically, in the early stage, ***EG*** is very large with nearly $0.8 \times$ NP elite particles. In this situation, the learning environments of particles in ***NEG*** are very different from each other and thus they can search for promising areas in different directions. As the evolution proceeds, ***EG*** becomes smaller and smaller, and in the late evolution stage, the size of ***EG*** becomes close to $0.4 \times$ NP. In this case, particles slightly tend to exploit the found optimal areas to refine the found solutions. However, it deserves mentioning that though the swarm biases to exploiting the found optimal regions as the evolution continues, the swarm diversity is not seriously sacrificed because during the evolution, ***EG*** always contains more than 0.4 × NP elite particles. With such many elites in ***EG***, the diversity of the learning environments for non-elite particles is still relatively high.

As for the consumption of memory, REELSO only needs to store the velocities and the positions of particles, which both take O(NP × D). Compared with traditional PSO variants based on historical evolutionary information (like ***pbest*** and ***gbest***), O(NP × D) space can be saved, because in REELSO, no historical evolutionary information needs to be stored.

Algorithm 1 The complete procedure of REELSO

```
Input: The swarm size NP, the maximum fitness evaluations FES_max, and the elite neighbor size ENS;
1:  Uniformly and randomly sample NP points in the solution space to initialize the swarm, calculate its fitness, and set fes = NP;
2:  While (fes ≤ FES_max) do
3:      Sort particles from the best to the worst in terms of their fitness;
4:      Calculate EGS according to Eq. (5);
5:      Divide the swarm into two groups: EG and NEG;
6:      For (each particle in NEG) do
7:          Randomly select ENS elites from EG;
8:          Find the best elite in the elite neighbor region;
9:          Update the particle according to Eq.(3) and Eq.(4);
10:         Evaluate the updated particle and fes++;
11:     End For
12: End While
13: Find the best particle x in the swarm.
Output: The best particle x and its fitness f(x);
```

In conclusion, with this adaptive partition strategy, the proposed REEL gradually changes from exploring the large-scale solution space to exploiting the found optimal zones subtly without serious sacrifice of swarm diversity. Such a property is very beneficial for PSO to explore the high-dimensional space and exploit the found promising zones appropriately and at the same time avoid falling into local areas.

Overall framework and complexity analysis

Combining the above two strategies together, we develop REELSO, whose pseudocode is outlined in Algorithm 1. From this algorithm, it is found that without consideration of the function evaluation time, the computing time of REELSO is O(NP × D) in each iteration, which is ineluctable for the update of particles. Specifically, it takes O(NP × logNP) to sort particles in the ascending order of their fitness as shown in Line 4, and takes O(NP) to separate particles into two groups as shown in Line 6 (actually in implementation this step can be saved). Then, it consumes O(NP × (ENS + ENS + ENS × D)) to update the particles in ***NEG*** as shown in Lines 7–12 (Line 8 and Line 9 take O(ENS) to construct the learning environment for each particle in ***NEG*** and find the best elite in the environment; Line 10 consumes O(ENS × D) to update each particle in ***NEG***). Since ENS is much smaller than NP, and both are usually smaller than D, the final computational time of REELSO in each generation is O(NP × D).

In brief, REELSO remains as efficient as classical PSOs in time consumption, but is more efficient in space occupation.

Difference between REELSO and existing large-scale PSOs

In the literature, some large-scale PSO variants also directly utilize superior particles in the current swarm to evolve inferior ones. To the best of our knowledge, CSO [47], SL-PSO [48], DLLSO [49], TPLSO [55], and SDLSO [56] are the most similar large-scale PSO variants to the proposed REELSO. In comparison with these five variants, REELSO distinguishes from them in the following aspects:

(1) REELSO randomly constructs an elite neighbor region for each member in ***NEG***. In particular, since the elites used to build the elite neighbor region are randomly chosen from ***EG***, the elite neighbor region is likely distinct for different members in ***NEG***. Besides, these elites afford positive learning for the non-elite particle and are all employed to direct the evolution of this particle. However, in the five large-scale PSO variants, each inferior particle is only guided by one or two superior ones. For example, in CSO, particles are paired together and the loser is evolved by learning from the winner, while the winner is not updated; in SL-PSO, each particle is triggered to update by a learning probability and once it is triggered to update, it only learns from a random superior particle; in DLLSO, TPLSO, and SDLSO, each

inferior particle is updated by two superior ones in the swarm. Based on the observational learning theory in [58–60], the surrounding environment of inferior particles in the five existing large-scale PSO variants is limited for them to observe and imitate. Therefore, inferior particles in REELSO are expected to preserve better learning ability than those in the five existing variants and thus REELSO is expected to achieve more promising optimization performance than the five large-scale PSO variants, which will be demonstrated by experiments in the later section.

(2) REELSO utilizes the best elite in the random elite neighbor region as the guiding exemplar to direct the cognitive learning and adopts all elites to guide the ensemble learning to update each particle in ***NEG***. From this perspective, the non-elite particles could acquire positive learning to approach promising areas fast and thus fast convergence could be guaranteed. In addition, since the elite neighbor region is distinct for different particles in ***NEG***, the exemplars in the cognitive learning and the ensemble learning are likely different as well for different non-elite particles. From this perspective, REELSO could preserve high diversity during the evolution. However, both CSO and SL-PSO utilize one random superior particle to direct the cognitive learning and the center of the swarm to direct the social learning to update inferior particles. Though the guiding exemplar in the cognitive learning is likely different for different inferior particles, the guiding exemplar in the social learning is the same for all inferior particles. From this respect, both CSO and SL-PSO preserve lower diversity in particle updating than REELSO. In addition, the exemplars in both the cognitive learning and the social learning are expectedly worse than those in REELSO. In this view of point, CSO and SL-PSO may preserve slower convergence than REELSO. Likewise, in DLLSO, TPLSO, and SDLSO, since they directly utilize two superior individuals in the current swarm to update inferior ones, there is no explicit social learning in these three PSO variants. Though the two exemplars are likely different for different inferior particles, they are expectedly worse than those in REELSO. Therefore, the learning ability of inferior particles in the three PSO variants is limited, leading to that they may preserve slower convergence than REELSO. Based on the above analysis, it is expected that REELSO could compromise high diversity and fast convergence better than these five existing large-scale PSO variants and thus REELSO is expected to obtain better optimization performance than the five PSO methods, which will be demonstrated by experiments in the later section as well.

Experiments

To demonstrate the feasibility and effectiveness of the devised REELSO, this section carries out abundant experiments on two public high-dimensional benchmark sets, namely the CEC'2010 [40] and the CEC'2013 [61] high-dimensional problem sets. The optimization problems in the CEC'2013 set are much harder to solve than those in the CEC'2010 set because they are generated by introducing more complex properties, such as imbalance and overlapping [61]. For more detailed information of these two sets, please refer to Refs. [40] and [61].

To comprehensively validate the effectiveness and efficiency of REELSO, we compare REELSO with 14 state-of-the-art optimizers designed for high-dimensional optimization. To be specific, the 14 state-of-the-art methods are TPLSO [55], SDLSO [56], DLLSO [49], CSO [47], SL-PSO [48], DECC-DG [50], DECC-XDG [74], DECC-GDG [75], DECC-DG2 [76], DECC-RDG [51], DECC-RDG2 [77], jDEsps [87], CO [88], and eWOA [89]. The former five algorithms and the last three methods are all non-decomposition large-scale optimizers proposed in recent years. However, the former five optimizers are all large-scale PSO variants, while the last three optimizers are the large-scale variants of other evolutionary algorithms, such as the differential evolution algorithm, the cheetah optimizer, and the whale optimization algorithm. The medium six methods are decomposition-based large-scale approaches. It should be mentioned that in the six decomposition-based methods [50, 51, 74–77], DE was utilized instead of PSO, because in the literature [50], DE has been experimentally demonstrated to be more promising than PSO in solving high-dimensional optimization problems under the decomposition frameworks. Besides, in the experiments, for fairness, the recommended settings (in the associated papers) of the parameters in the compared methods are directly adopted.

In the experiments, without otherwise stated, we set the maximum number of function evaluations as $3000 \times D$ (D denotes the dimension size) for all algorithms. For fair and comprehensive comparisons, this paper runs each algorithm independently 30 times, and then utilizes the median value, the mean value, and the standard deviation (Std) value over the 30 independent runs to measure the optimization performance of each method.

Furthermore, during the compassions between REELSO and the 14 compared large-scale methods, the Wilcoxon rank sum test at the significance level of $\alpha = 0.05$ is performed to tell whether there is significant difference between the optimization result of the proposed REELSO and that of each compared method on each optimization problem. After the execution, the p value is output. If the p value is larger than 0.05, the devised REELSO performs equivalently with the associated compared method on the corresponding optimization problem. Otherwise, there is significant difference between the optimization result of REELSO and that of the associated compared method. Based on this principle, in the following tables, the mark "+" above the p values implies that REELSO significantly outperforms the corresponding compared methods, and "−" means that REELSO is significantly inferior to the associated compared methods, while "=" implies that REELSO achieves equivalent performance with the corresponding compared methods. Accordingly, "*w/t/l*" count the numbers of "+", "=" and "−", respectively. Besides, the Friedman test at the significance level of $\alpha = 0.05$ is performed to acquire the overall ranks of all methods on one whole benchmark set, so that the overall optimization performance of all methods can be compared. After the execution, the average rank of each algorithm is output. In particular, the smaller the rank value of one algorithm is, the better overall optimization performance the algorithm attains.

Lastly, it deserves mentioning that we run all experiments on a PC with 8 Intel Core i7-10700 2.90-GHz CPUs, 8-GB memory and the 64-bit Ubuntu 12.04 LTS system.

Investigation of REELSO

1. *Parameter settings* In REELSO, three parameters need special fine-tuning, that is, the swarm size NP, the parameter ϕ in Eq. (3), and the elite neighbor size ENS. The swarm size NP is a common parameter of all PSOs and it is usually problem-dependent. The elite neighbor size ENS determines the size of the learning environment of each particle in the non-elite group. Specifically, as ENS increases, more and more elites are involved in the elite neighbor region of each particle in the non-elite group. As a result, the attraction of the best elite in the cognitive learning becomes greedier and greedier and more and more elites participate in the ensemble learning. Therefore, with ENS increasing, the swarm gradually biases to exploiting the solution space at the risk of losing the swarm diversity. Hence, to properly explore and exploit the search space, such a parameter needs fine-tuning for REELSO to obtain satisfactory performance. As for the control parameter ϕ, it takes charge of the influence of the ensemble learning. A large ϕ enhances the influence of the ensemble learning part. In this situation, the attraction of the best elite in the cognitive learning part could be weakened, and thus the swarm diversity could be improved. However, this may slow down the convergence of the swarm, which is not profitable for the swarm to promote the accuracy of the found solutions. By contrast, a small ϕ declines the impact of the ensemble learning part. In this case, the updated particle obtains more observation from the best elite in the cognitive learning part, which is profitable for it to quickly approach the area where the best elite lies. However, once the best elite falls into a local area, the updated particle may also fall into the local area. Therefore, the control parameter ϕ needs to be set properly for REELSO to search the space appropriately to find high-quality solutions. Based on the above analysis, it is found that ENS and ϕ may interfere with the proper setting of each other because both of them make direct influence on the ensemble learning. Therefore, in the following, we first investigate the optimal setting of NP and then seek the optimal settings of ENS and ϕ simultaneously.

First, to investigate the proper setting of NP, we perform experiments on the CEC'2010 set with NP varying from 400 to 1000. Table S1 in the supplementary material displays the comparison results among REELSO with different settings of NP. In this table, the average rank of each setting is shown in the last row by conducting the Friedman test on the whole 20 problems. Besides, the best optimization results are also bolded in the table.

From Table S1, the following observations can be attained. (1) From the perspective of the average rank obtained from the Friedman test, REELSO with NP = 800 achieves the lowest rank. This indicates that such a setting of NP helps REELSO perform the best over the whole 1000-*D* CEC'2010 benchmark problem set. (2) Specifically, with NP = 800, REELSO obtains the best performance on 8 problems, while with the other settings of NP, it achieves the best results on at most 6 problems. In particular, taking deep comparison between REELSO with NP = 800 and the ones with the other settings, we find that on the other 12 problems, the difference between the optimization results obtained by REELSO with NP = 800 and those obtained by REELSO with the associated optimal NP is very small. Based on these observations, this paper sets the swarm size NP as 800 for REELSO to solve 1000-*D* problems.

Subsequently, to investigate the optimal settings of the elite neighbor size ENS and the control parameter ϕ simultaneously, this paper carries out experiments on the 1000-*D* CEC'2010 optimization problem set with ϕ varying from 0.05 to 0.30 and ENS varying from 6 to 11. Table S2 in the supplementary material displays the comparison results among REELSO with different configurations of these two parameters. Specifically, in this table, for each setting of ENS, the average optimization results of REELSO with different settings of ϕ over 30 independent runs on each problem are reported. In particular, for each setting of ENS, the best results obtained by REELSO with the optimal setting of ϕ are highlighted in bold. To observe the overall optimization performance of REELSO with each combination of ENS and ϕ, the Friedman test is conducted over all optimization results obtained by REELSO with all combinations of ENS and ϕ at the significance level of $\alpha = 0.05$.

Taking deep observation on Table S2, we can attain the following findings:

- As displayed in the last row of each part, in view of the average rank, it is found that REELSO with ENS $= 9$ and $\phi = 0.10$ achieves the lowest rank. This indicates that REELSO with such a combination of ENS and ϕ obtains the best overall optimization performance among all configurations of ENS and ϕ on the whole CEC'2010 benchmark set.
- For each setting of ENS, it is interesting to find that no matter with respect to the average rank or from the view point of the number of the problems where REELSO acquires the best performance, REELSO with $\phi = 0.10$ achieves much better performance than the ones with the other settings of ϕ. In particular, it is found that when ϕ exceeds 0.10 or is lower than 0.10, the performance of REELSO sharply deteriorates no matter what ENS is. Therefore, we keep $\phi = 0.10$ for REELSO to solve any optimization problems.
- As for ENS, it is found that with $\phi = 0.10$, when ENS is too large, such as ENS $= 11$, or ENS is too small, such as ENS $= 6$, the optimization performance of REELSO degrades. This is because a too large ENS or a too small ENS could not help REELSO compromise the diversity and the convergence well to search the large-scale space. Based on the average rank, this paper keeps ENS $= 9$ for REELSO to solve any optimization problems.

To summarize, based on the above investigation experiments, NP $= 800$, ENS $= 9$ and $\phi = 0.10$ are adopted for REELSO to solve 1000-D problems.

2. *Influence of the adaptive partition strategy* To further help REELSO achieve a good compromise between search diversification and intensification, this paper devises an adaptive partition strategy (as shown in Eq. (5)) by dynamically adjusting the size of the elite group, namely EGS. In this strategy, three parameters are involved, namely, EGS_{min}, EGS_{max} and α. Therefore, we first conduct experiments on the CEC'2010 benchmark set to investigate the appropriate settings of these parameters before the verification of the effectiveness of this adaptive strategy.

Firstly, to investigate the appropriate range of EGS, 12 different combinations of EGS_{min} and EGS_{max} are configured for REELSO with EGS_{min} varying from 0.3 to 0.6 and EGS_{max} varying from 0.7 to 0.9. The experimental results of REELSO with different configurations of EGS_{min} and EGS_{max} on the CEC'2010 set are shown in Table S3 in the supplementary material. In this table, the best results are bolded and the average rank of each configuration attained from the Friedman test is shown in the last row.

With careful observation on Table S3, from the perspective of the average rank, it is found that REELSO with $EGS_{min} = 0.4$ and $EGS_{max} = 0.8$ achieves the lowest rank and such a rank is much smaller than those of REELSO with the other settings of EGS_{min} and EGS_{max}. This implies that REELSO with such a combination of EGS_{min} and EGS_{max} obtains the best overall optimization performance among all combinations of EGS_{min} and EGS_{max} and such a combination shows significant superiority to other combinations. Based on these observations, this paper sets $EGS_{min} = 0.4$ and $EGS_{max} = 0.8$ for REELSO to solve high-dimensional optimization problems.

Subsequently, to investigate the optimal setting of the parameter α, this paper carries out experiments with α varying from 0.1 to 0.9. Table S4 in the supplementary material displays the comparison results among REELSO with different configurations of α on the CEC'2010 benchmark set.

From Table S4, it is found that REELSO with $\alpha = 0.8$ achieves the lowest rank and such a rank is much smaller than those of REELSO with the other settings of α. This shows that REELSO with such a setting of α obtains the best overall optimization performance among all settings of α on the whole CEC'2010 benchmark set.

To summarize, based on the above investigation experiments, $EGS_{min} = 0.4$, $EGS_{max} = 0.8$ and $\alpha = 0.8$ are adopted in Eq. (5) for REELSO to solve high-dimensional optimization problems.

Subsequently, to testify the usefulness of the devised adaptive strategy, this paper executes experiments on the CEC'2010 benchmark problems to compare REELSO with this adaptive strategy and those with different fixed EGS. Specifically, six different fixed settings of EGS are adopted, namely EGS = 0.4*NP, EGS = 0.5*NP, EGS = 0.6*NP, EGS = 0.7*NP, EGS = 0.8*NP and EGS = 0.9*NP. Table 1 presents the experimental results of REELSO with the adaptive strategy and different fixed EGS on the CEC'2010 benchmark problems.

Taking a close look at Table 1, we acquire the following findings:

(1) From the perspective of the average rank attained from the Friedman test, it is found that REELSO with the adaptive partition strategy achieves the lowest rank and such a rank is much smaller than those of REELSO with the six fixed settings of EGS. This indicates that REELSO with the adaptive partition scheme performs the best among all versions of REELSO over all the twenty 1000-D CEC'2010 problems.

Table 1 Comparison results between REELSO with and without the adaptive strategy on the 1000-*D* CEC'2010 benchmark problems

F	EGS						
	Adaptive EGS	EGS = 0.4*NP	EGS = 0.5*NP	EGS = 0.6*NP	EGS = 0.7*NP	EGS = 0.8*NP	EGS = 0.9*NP
F_1	**1.36E−24**	1.44E−24	2.63E−24	3.53E−24	5.51E−24	7.27E−24	7.64E−24
F_2	1.22E+03	1.54E+03	1.39E+03	1.33E+03	1.27E+03	1.24E+03	**1.18E+03**
F_3	1.99E−14	2.02E−14	1.99E−14	1.81E−14	**1.71E−14**	1.77E−14	1.82E−14
F_4	**5.27E+10**	5.69E+10	5.70E+10	5.51E+10	5.56E+10	5.92E+10	5.67E+10
F_5	**6.84E+06**	9.96E+06	9.32E+06	7.80E+06	7.50E+06	1.57E+07	2.44E+07
F_6	1.69E+01	1.97E+01	1.97E+01	1.96E+01	1.98E+01	1.89E+01	**1.12E+01**
F_7	**3.10E−12**	1.38E+00	3.83E−02	3.01E−01	1.24E−10	2.90E−11	4.63E−11
F_8	1.40E+04	1.41E+04	1.24E+04	1.27E+04	1.40E+04	1.53E+04	**1.23E+04**
F_9	6.71E+06	6.97E+06	**5.92E+06**	6.01E+06	6.12E+06	6.43E+06	6.58E+06
F_{10}	1.29E+03	1.60E+03	1.48E+03	1.40E+03	1.34E+03	1.29E+03	**1.23E+03**
F_{11}	**1.99E+01**	2.12E+01	2.02E+01	2.02E+01	2.05E+01	2.02E+01	2.05E+01
F_{12}	**5.05E+01**	1.91E+02	8.19E+01	8.38E+01	8.71E+01	8.19E+01	8.23E+01
F_{13}	2.07E+02	2.67E+02	2.06E+02	**1.80E+02**	1.99E+02	2.27E+02	2.42E+02
F_{14}	1.68E+07	1.91E+07	1.66E+07	**1.62E+07**	1.70E+07	1.74E+07	1.87E+07
F_{15}	**1.34E+03**	1.69E+03	1.54E+03	1.44E+03	1.36E+03	2.83E+03	6.16E+03
F_{16}	5.04E+00	1.09E+01	6.76E+00	6.17E+00	6.35E+00	**3.85E+00**	4.26E+00
F_{17}	1.54E+03	**1.43E+03**	1.65E+03	1.73E+03	2.44E+03	2.90E+03	4.17E+03
F_{18}	**7.07E+02**	9.05E+02	7.40E+02	7.29E+02	7.55E+02	7.43E+02	8.29E+02
F_{19}	1.35E+06	**7.94E+05**	9.49E+05	1.30E+06	2.51E+06	3.87E+06	4.77E+06
F_{20}	**8.93E+02**	1.01E+03	9.36E+02	9.25E+02	9.06E+02	9.04E+02	8.97E+02
Rank	**2.38**	5.73	3.93	3.40	4.05	4.20	4.33

The best results are highlighted in bold

(2) In-depth investigation on the comparison results demonstrates that the adaptive strategy helps REELSO achieve the best optimization performance on nine benchmark problems, while REELSO with the fixed settings of EGS performs the best on no more than four problems. Besides, it is also found that on the other 11 problems, where REELSO with the adaptive strategy obtains inferior performance, the difference between the optimization results obtained by REELSO with the adaptive strategy and those obtained by REELSO with the associated optimal settings of EGS is very small.

All in all, according to the above findings, the designed adaptive partition scheme is very profitable for REELSO to attain promising performance in solving high-dimensional problems.

Comparison with state-of-the-art methods

This section conducts experiments on the CEC'2010 and the CEC'2013 benchmark sets to compare REELSO with the 14 compared large-scale optimizers. Table 2 shows the summarized statistical comparison results between REELSO and the 14 compared methods on different types of benchmark problems in the two benchmark sets, while Tables 3 and 4 show the detailed experimental results on the 1000-*D* CEC'2010 and the 1000-*D* CEC'2013 benchmark sets, respectively.

From Tables 2 and 3, we can obtain the following findings on the twenty 1000-*D* CEC'2010 benchmark problems:

(1) From the perspective of the average rank achieved from the Friedman test, it is found that REELSO achieves the lowest rank among the 15 algorithms. This implies that REELSO performs the best over the whole 1000-*D* CEC'2010 benchmark set.
(2) With respect to "*w/t/l*" counted on the basis of the Wilcoxon rank sum test, REELSO significantly outperforms the 14 compared methods on more than 13 problems, and only displays inferiority to them on no more than 6 problems. In particular, compared with the five non-decomposition-based large-scale PSOs, namely TPLSO, SDLSO, DLLSO, CSO, and SL-PSO, REELSO exhibits significant superiority to them on 16, 13, 14, 16, and 19 problems, respectively. As compared to the six state-of-the-art decomposition-based

Table 2 Summarized statistical results between REELSO and the compared methods on the CEC'2010 and CEC'2013 benchmark sets

Benchmark set	Problem property	Index	REEPSO	TPLSO	SDLSO	DLLSO	CSO	SL-PSO	DECC-DG	DECC-XDG	DECC-GDG	DECC-DG2	DECC-RDG	DECC-RDG2	jDEsps	CO	eWOA
CEC2010-1000	Fully separable unimodal	*w/t/l*	–	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0
	Fully separable multi-modal	*w/t/l*	–	1/0/1	1/0/1	1/0/1	2/0/0	2/0/0	2/0/0	2/0/0	1/0/1	2/0/0	2/0/0	2/0/0	1/0/1	1/0/1	2/0/0
	Partially separable unimodal	*w/t/l*	–	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0
	Partially separable multi-modal	*w/t/l*	–	6/0/3	3/2/4	4/1/4	5/1/3	8/0/1	6/1/2	6/1/2	7/1/1	6/1/2	6/1/2	6/1/2	5/0/4	9/0/0	9/0/0
	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	1/0/0	1/0/0	1/0/0	0/0/1	0/0/1	1/0/0
	Fully non-separable multi-modal	*w/t/l*	–	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	0/1/0	1/0/0	1/0/0
	Overall	*w/t/l*	–	16/0/4	13/2/5	14/1/5	16/1/3	19/0/1	17/1/2	17/1/2	17/1/2	17/1/2	17/1/2	17/1/2	13/1/6	18/0/2	20/0/0
	Overall	*Rank*	**3.40**	7.05	3.50	5.70	8.60	10.30	9.65	10.60	10.20	9.15	6.80	7.20	5.25	7.60	15.00

Table 2 (continued)

Benchmark set	Problem property	Index	REEPSO	TPLSO	SDLSO	DLLSO	CSO	SL-PSO	DECC-DG	DECC-XDG	DECC-GDG	DECC-DG2	DECC-RDG	DECC-RDG2	jDEsps	CO	eWOA
CEC2013-1000	Fully separable unimodal	*w/t/l*	–	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0
	Fully separable multi-modal	*w/t/l*	–	1/0/1	1/0/1	1/0/1	2/0/0	2/0/0	2/0/0	2/0/0	1/0/1	2/0/0	2/0/0	2/0/0	1/0/1	1/0/1	2/0/0
	Partially separable unimodal	*w/t/l*	–	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	3/0/0	2/1/0	3/0/0	3/0/0
	Partially separable multi-modal	*w/t/l*	–	3/2/0	2/3/0	3/2/0	4/1/0	3/2/0	4/1/0	4/1/0	5/0/0	4/1/0	5/0/0	4/1/0	3/0/2	2/0/3	3/0/2
	Overlapping unimodal	*w/t/l*	–	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0
	Overlapping multi-modal	*w/t/l*	–	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	0/1/0	1/0/0	1/0/0
	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	1/0/0	1/0/0	1/0/0	1/0/0	0/0/1	1/0/0
	Overall	*w/t/l*	–	11/3/1	10/4/1	11/3/1	13/2/0	12/3/0	13/1/1	13/1/1	13/0/2	13/1/1	14/0/1	13/1/1	9/2/4	9/0/6	12/1/2
	Overall	*Rank*	**3.13**	5.33	5.93	6.00	7.53	8.40	11.07	9.20	10.60	10.80	9.73	8.53	6.67	4.00	13.07

Bold values mean that the associated algorithm achieves the lowest average rank over the whole benchmark set

Table 3 Performance comparison between REELSO and the 14 compared large-scale optimizers on the 1000-*D* CEC'2010 problems

F	Quality	REELSO	TPLSO	SDLSO	DLLSO	CSO	SL-PSO	DECC-DG	DECC-XDG
F_1	Median	1.21E−24	1.94E−18	2.44E−23	2.80E−22	4.63E−12	7.51E−18	3.88E+02	6.40E+02
	Mean	1.36E−24	1.88E−18	2.76E−23	3.00E−22	4.75E−12	5.75E+01	7.34E+03	5.76E+04
	Std	7.69E−25	3.02E−19	1.07E−23	7.15E−23	7.90E−13	1.82E+02	1.87E+04	2.42E+05
	p value	–	**6.47E−08^{+}**	**6.47E−08^{+}**	**5.51E−09^{+}**	**6.47E−08^{+}**	**3.54E−05^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**
F_2	Median	1.19E+03	1.07E+03	9.06E+02	9.68E+02	7.51E+03	1.85E+03	4.36E+03	4.47E+03
	Mean	1.22E+03	1.07E+03	9.09E+02	9.79E+02	7.48E+03	1.85E+03	4.37E+03	4.44E+03
	Std	5.87E+01	6.80E+01	3.88E+01	5.53E+01	2.63E+02	1.61E+01	1.64E+02	1.99E+02
	p value	–	3.11E−07^{-}	6.46E−08^{-}	5.50E−09^{-}	**6.46E−08^{+}**	**2.31E−05^{+}**	**6.46E−08^{+}**	**6.46E−08^{+}**
F_3	Median	2.18E−14	1.40E+00	2.53E−14	2.89E−14	2.56E−09	2.01E+00	1.67E+01	1.67E+01
	Mean	1.99E−14	1.41E+00	2.52E−14	2.73E−14	2.57E−09	2.01E+00	1.67E+01	1.67E+01
	Std	2.96E−15	1.25E−01	1.74E−15	2.40E−15	1.85E−10	6.32E−06	2.94E−01	2.48E−01
	p value	–	**5.45E−08^{+}**	**7.25E−08^{+}**	**3.63E−09^{+}**	**5.45E−08^{+}**	**1.16E−05^{+}**	**5.45E−08^{+}**	**5.45E−08^{+}**
F_4	Median	4.90E+10	2.72E+11	1.38E+11	4.35E+11	6.92E+11	2.95E+11	5.27E+12	6.92E+11
	Mean	5.27E+10	2.85E+11	1.42E+11	4.43E+11	6.87E+11	5.25E+11	5.20E+12	7.45E+11
	Std	8.43E+09	9.32E+10	3.21E+10	1.19E+11	1.79E+11	5.37E+11	1.70E+12	2.65E+11
	p value	–	**6.47E−08^{+}**	**6.47E−08^{+}**	**5.51E−09^{+}**	**6.47E−08^{+}**	**3.59E−05^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**
F_5	Median	6.97E+06	1.54E+07	7.96E+06	1.09E+07	2.00E+06	2.59E+07	1.50E+08	1.57E+08
	Mean	6.84E+06	1.56E+07	7.99E+06	1.14E+07	2.46E+06	2.59E+07	1.56E+08	1.57E+08
	Std	2.02E+06	4.53E+06	2.10E+06	2.62E+06	1.35E+06	1.26E+02	1.98E+07	2.45E+07
	p value	–	**2.74E−07^{+}**	3.42E−01$^{=}$	**3.90E−06^{+}**	9.60E−07^{-}	**2.32E−05^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**
F_6	Median	1.93E+01	2.05E+00	4.00E−09	4.00E−09	8.18E−07	3.53E+00	1.63E+01	1.64E+01
	Mean	1.69E+01	2.17E+00	4.00E−09	4.00E−09	8.16E−07	4.99E+00	1.63E+01	1.63E+01
	Std	6.46E+00	3.77E−01	8.41E−25	3.47E−15	2.60E−08	4.05E+00	3.44E−01	3.23E−01
	p value	–	7.47E−05^{-}	1.17E−10^{-}	1.64E−10^{-}	6.47E−08^{-}	2.12E−03^{-}	7.47E−05^{-}	7.47E−05^{-}
F_7	Median	4.70E−13	9.00E+02	6.62E−02	7.82E+00	2.13E+04	1.25E+05	7.14E+03	3.67E+02
	Mean	3.10E−12	5.73E+03	5.80E−01	4.21E+01	2.13E+04	5.33E+05	9.77E+03	1.31E+03
	Std	9.02E−12	1.03E+04	2.78E+00	1.53E+02	4.60E+03	1.10E+06	7.54E+03	2.11E+03
	p value	–	**6.47E−08^{+}**	**6.47E−08^{+}**	**5.51E−09^{+}**	**6.47E−08^{+}**	**3.59E−05^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**
F_8	Median	1.22E+04	4.74E+05	2.32E+04	2.33E+07	3.86E+07	1.13E+07	1.66E+07	1.50E+01
	Mean	1.40E+04	4.93E+05	2.36E+04	2.33E+07	3.86E+07	1.49E+07	2.39E+07	3.99E+05
	Std	5.17E+03	1.46E+05	2.97E+03	2.56E+05	8.33E+04	1.55E+07	1.95E+07	1.22E+06
	p value	–	**6.47E−08^{+}**	**5.66E−06^{+}**	**5.51E−09^{+}**	**6.47E−08^{+}**	**3.59E−05^{+}**	**6.47E−08^{+}**	**1.55E−05^{+}**
F_9	Median	6.72E+06	4.20E+07	2.29E+07	4.55E+07	6.64E+07	3.76E+07	5.70E+07	1.12E+08
	Mean	6.71E+06	4.28E+07	2.30E+07	4.49E+07	6.68E+07	5.97E+07	5.76E+07	1.15E+08
	Std	5.22E+05	4.17E+06	2.24E+06	3.57E+06	4.47E+06	5.70E+07	8.26E+06	1.48E+07
	p value	–	**6.47E−08^{+}**	**6.47E−08^{+}**	**5.51E−09^{+}**	**6.47E−08^{+}**	**3.59E−05^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**
F_{10}	Median	1.30E+03	9.60E+02	8.26E+02	8.88E+02	9.58E+03	3.00E+03	4.48E+03	5.21E+03
	Mean	1.29E+03	9.68E+02	8.33E+02	8.86E+02	9.58E+03	5.27E+03	4.49E+03	5.15E+03
	Std	5.56E+01	6.88E+01	4.17E+01	3.90E+01	6.68E+01	3.30E+03	1.31E+02	1.35E+02
	p value	–	6.47E−08^{-}	6.45E−08^{-}	5.50E−09^{-}	**6.47E−08^{+}**	**3.47E−05^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**
F_{11}	Median	1.99E+01	3.44E+00	1.43E−13	2.36E+00	3.97E−08	2.08E+01	1.03E+01	1.09E+01
	Mean	1.99E+01	3.45E+00	1.43E−13	4.51E+00	3.98E−08	2.09E+01	1.03E+01	1.07E+01
	Std	1.21E−01	1.33E+00	4.79E−15	4.83E+00	3.25E−09	5.22E−01	9.23E−01	9.09E−01
	p value	-	6.47E−08^{-}	4.03E−08^{-}	7.93E−09^{-}	6.47E−08^{-}	**3.59E−05^{+}**	6.47E−08^{-}	6.47E−08^{-}

Table 3 (continued)

F	Quality	REELSO	TPLSO	SDLSO	DLLSO	CSO	SL-PSO	DECC-DG	DECC-XDG
F_{12}	Median	3.59E+01	1.17E+04	5.90E+03	1.23E+04	4.25E+05	1.77E+04	2.36E+03	1.28E+04
	Mean	5.05E+01	1.18E+04	5.93E+03	1.22E+04	4.37E+05	6.39E+04	2.85E+03	1.25E+04
	Std	4.71E+01	1.26E+03	6.23E+02	1.26E+03	6.61E+04	1.09E+05	1.01E+03	2.10E+03
	p value	-	**6.47E−08$^+$**	**6.47E−08$^+$**	**5.51E−09$^+$**	**6.47E−08$^+$**	**3.59E−05$^+$**	**6.47E−08$^+$**	**6.47E−08$^+$**
F_{13}	Median	1.77E+02	7.22E+02	4.81E+02	7.81E+02	4.68E+02	9.74E+02	4.24E+03	2.85E+03
	Mean	2.07E+02	7.49E+02	5.30E+02	8.18E+02	5.53E+02	1.53E+03	5.58E+03	3.21E+03
	Std	7.52E+01	1.09E+02	1.53E+02	2.64E+02	1.78E+02	9.01E+02	3.40E+03	1.04E+03
	p value	-	**6.47E−08$^+$**	**7.40E−08$^+$**	**5.51E−09$^+$**	**6.47E−08$^+$**	**3.59E−05$^+$**	**6.47E−08$^+$**	**6.47E−08$^+$**
F_{14}	Median	1.67E+07	1.27E+08	6.50E+07	1.22E+08	2.46E+08	8.07E+07	3.43E+08	5.94E+08
	Mean	1.68E+07	1.27E+08	6.61E+07	1.22E+08	2.46E+08	1.29E+08	3.42E+08	5.96E+08
	Std	9.95E+05	8.25E+06	4.07E+06	7.91E+06	1.31E+07	1.26E+08	2.30E+07	3.14E+07
	p value	-	**6.47E−08$^+$**	**6.47E−08$^+$**	**5.51E−09$^+$**	**6.47E−08$^+$**	**3.59E−05$^+$**	**6.47E−08$^+$**	**6.47E−08$^+$**
F_{15}	Median	1.33E+03	1.01E+04	5.98E+02	8.32E+02	1.01E+04	1.13E+04	5.89E+03	6.34E+03
	Mean	1.34E+03	8.56E+03	9.00E+02	8.71E+02	1.01E+04	1.13E+04	5.87E+03	6.37E+03
	Std	4.92E+01	3.33E+03	1.66E+03	2.75E+02	5.84E+01	2.48E+01	1.01E+02	9.90E+01
	p value	-	**1.55E−05$^+$**	4.54E−07$^−$	2.17E−08$^−$	**6.46E−08$^+$**	**2.32E−05$^+$**	**6.47E−08$^+$**	**6.47E−08$^+$**
F_{16}	Median	2.08E+00	1.72E+01	1.94E−13	3.98E+00	5.64E−08	1.04E+01	7.37E−13	1.74E−08
	Mean	5.04E+00	1.86E+01	1.94E−13	4.20E+00	5.68E−08	1.08E+01	7.33E−13	1.79E−08
	Std	8.50E+00	7.44E+00	4.61E−15	2.11E+00	6.32E−09	1.20E+00	4.09E−14	1.77E−09
	p value	-	**1.55E−05$^+$**	2.80E−01$^=$	5.12E−02$^=$	2.84E−01$^=$	**2.11E−03$^+$**	2.84E−01$^=$	2.84E−01$^=$
F_{17}	Median	1.42E+03	9.61E+04	9.23E+04	9.14E+04	2.18E+06	9.28E+04	4.11E+04	1.24E+05
	Mean	1.54E+03	9.64E+04	9.19E+04	9.15E+04	2.21E+06	2.37E+05	4.11E+04	1.24E+05
	Std	4.27E+02	8.15E+03	3.83E+03	5.14E+03	2.10E+05	3.32E+05	2.45E+03	6.58E+03
	p value	-	**6.47E−08$^+$**	**6.47E−08$^+$**	**5.51E−09$^+$**	**6.47E−08$^+$**	**3.59E−05$^+$**	**6.47E−08$^+$**	**6.47E−08$^+$**
F_{18}	Median	6.90E+02	2.27E+03	1.26E+03	2.51E+03	1.38E+03	3.73E+03	1.46E+10	1.39E+03
	Mean	7.07E+02	2.31E+03	1.31E+03	2.55E+03	1.64E+03	5.43E+03	1.48E+10	1.37E+03
	Std	1.37E+02	4.27E+02	2.43E+02	7.25E+02	8.27E+02	4.88E+03	2.45E+09	1.55E+02
	p value	-	**6.47E−08$^+$**	**6.47E−08$^+$**	**5.51E−09$^+$**	**6.47E−08$^+$**	**3.59E−05$^+$**	**6.47E−08$^+$**	**6.47E−08$^+$**
F_{19}	Median	1.33E+06	3.88E+06	4.86E+06	1.85E+06	9.78E+06	7.36E+06	1.76E+06	1.71E+06
	Mean	1.35E+06	3.86E+06	4.86E+06	1.83E+06	9.86E+06	9.10E+06	1.74E+06	1.73E+06
	Std	7.37E+04	2.76E+05	2.81E+05	9.21E+04	5.13E+05	4.56E+06	1.02E+05	7.02E+04
	p value	-	**6.47E−08$^+$**	**6.47E−08$^+$**	**5.51E−09$^+$**	**6.47E−08$^+$**	**3.59E−05$^+$**	**6.47E−08$^+$**	**6.47E−08$^+$**
F_{20}	Median	8.87E+02	2.00E+03	1.23E+03	1.90E+03	9.87E+02	1.93E+03	6.39E+10	2.04E+04
	Mean	8.93E+02	2.04E+03	1.22E+03	1.92E+03	1.07E+03	2.14E+03	6.45E+10	2.19E+06
	Std	1.81E+01	1.89E+02	1.16E+02	2.62E+02	1.72E+02	6.20E+02	8.66E+09	9.79E+06
	p value	-	**6.47E−08$^+$**	**6.47E−08$^+$**	**5.51E−09$^+$**	**6.47E−08$^+$**	**3.59E−05$^+$**	**6.47E−08$^+$**	**6.47E−08$^+$**
w/t/l		-	16/0/4	13/2/5	14/1/5	16/1/3	19/0/1	17/1/2	17/1/2
Rank		**3.40**	7.05	3.50	5.70	8.60	10.30	9.65	10.60

F	Quality	DECC-GDG	DECC-DG2	DECC-RDG	DECC-RDG2	jDEsps	CO	eWOA
F_1	Median	3.42E−10	3.49E+00	5.73E−01	1.39E−01	6.48E−21	3.99E−01	1.73E+11
	Mean	3.89E−10	2.23E+01	3.10E+01	2.85E+01	2.01E−16	4.04E−01	1.73E+11
	Std	1.37E−10	7.08E+01	1.20E+02	1.15E+02	9.09E−16	6.37E−02	7.14E+09
	p value	**6.47E−08$^+$**	**6.47E−08$^+$**	**6.47E−08$^+$**	**6.47E−08$^+$**	**2.14E−05$^+$**	**3.59E−05$^+$**	**5.44E−08$^+$**
F_2	Median	4.98E+02	4.45E+03	4.42E+03	4.40E+03	4.38E+01	9.07E+01	1.62E+04

Table 3 (continued)

F	Quality	DECC-GDG	DECC-DG2	DECC-RDG	DECC-RDG2	jDEsps	CO	eWOA
	Mean	5.00E+02	4.43E+03	4.39E+03	4.40E+03	6.87E+01	9.55E+01	1.62E+04
	Std	1.92E+01	1.89E+02	1.94E+02	1.45E+02	7.22E+01	8.93E+00	1.07E+02
	p value	6.46E−08^{-}	**6.46E−08^{+}**	**6.46E−08^{+}**	**6.45E−08^{+}**	6.30E−08^{-}	3.58E−05^{-}	**1.23E−03^{+}**
F_3	Median	1.68E+01	1.67E+01	1.67E+01	1.66E+01	2.62E−13	3.39E−04	2.09E+01
	Mean	1.67E+01	1.67E+01	1.67E+01	1.66E+01	6.13E−12	3.34E−04	2.09E+01
	Std	3.27E−01	2.80E−01	3.01E−01	2.89E−01	3.18E−11	1.39E−05	1.62E−02
	p value	**5.45E−08^{+}**	**5.45E−08^{+}**	**5.45E−08^{+}**	**5.45E−08^{+}**	**5.45E−08^{+}**	**1.91E−05^{+}**	**5.45E−08^{+}**
F_4	Median	7.00E+13	8.68E+11	7.11E+11	6.58E+11	1.90E+11	1.34E+11	1.32E+15
	Mean	6.83E+13	8.49E+11	7.58E+11	7.03E+11	4.98E+11	1.43E+11	1.22E+15
	Std	1.14E+13	2.61E+11	2.40E+11	2.69E+11	1.06E+12	4.04E+10	4.45E+14
	p value	**6.47E−08^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**	**3.59E−05^{+}**	**6.47E−08^{+}**
F_5	Median	3.58E+08	1.37E+08	1.27E+08	1.27E+08	7.07E+07	2.58E+08	6.95E+08
	Mean	3.53E+08	1.39E+08	1.28E+08	1.29E+08	6.99E+07	2.89E+08	6.83E+08
	Std	1.98E+07	2.42E+07	2.17E+07	2.13E+07	1.69E+07	1.06E+08	4.66E+07
	p value	**6.47E−08^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**	**3.59E−05^{+}**	**6.47E−08^{+}**
F_6	Median	3.44E+02	1.53E+01	1.61E+01	1.63E+01	4.27E−09	8.66E+06	2.02E+07
	Mean	3.63E+02	1.53E+01	1.61E+01	1.63E+01	5.95E−04	1.04E+07	2.02E+07
	Std	9.17E+01	3.06E−01	3.29E−01	4.09E−01	3.14E−03	4.91E+06	2.22E+05
	p value	**6.47E−08^{+}**	7.47E−05^{-}	7.47E−05^{-}	7.47E−05^{-}	6.47E−08^{-}	**3.59E−05^{+}**	**6.47E−08^{+}**
F_7	Median	1.71E+10	2.81E+01	1.03E+01	4.41E+00	1.88E+04	3.19E+00	4.88E+11
	Mean	1.73E+10	3.61E+02	8.47E+01	9.29E+01	6.39E+05	3.20E+00	4.84E+11
	Std	2.78E+09	7.77E+02	2.88E+02	2.32E+02	2.10E+06	7.41E−01	1.48E+11
	p value	**6.47E−08^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**	**3.59E−05^{+}**	**6.47E−08^{+}**
F_8	Median	1.00E+08	5.22E+01	7.50E+00	6.13E+00	8.04E−12	4.84E+06	3.14E+16
	Mean	1.26E+08	6.65E+05	2.66E+05	1.33E+05	1.75E+00	2.69E+07	3.02E+16
	Std	1.15E+08	1.51E+06	1.01E+06	7.28E+05	9.59E+00	3.45E+07	7.45E+15
	p value	**6.47E−08^{+}**	**3.19E−04^{+}**	**2.83E−06^{+}**	**4.56E−07^{+}**	6.47E−08^{-}	**4.28E−03^{+}**	**6.47E−08^{+}**
F_9	Median	3.87E+08	6.93E+07	4.65E+07	4.80E+07	9.29E+06	2.43E+07	2.05E+11
	Mean	3.90E+08	7.19E+07	4.75E+07	4.75E+07	2.78E+07	2.44E+07	2.04E+11
	Std	2.27E+07	1.07E+07	9.58E+06	5.51E+06	6.08E+07	1.74E+06	8.33E+09
	p value	**6.47E−08^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**	**2.32E−03^{+}**	**3.59E−05^{+}**	**2.80E−07^{+}**
F_{10}	Median	3.40E+03	4.68E+03	4.38E+03	4.37E+03	3.06E+03	5.08E+03	1.65E+04
	Mean	3.40E+03	4.69E+03	4.33E+03	4.34E+03	3.04E+03	5.10E+03	1.65E+04
	Std	7.24E+01	1.49E+02	1.39E+02	1.08E+02	4.07E+02	2.65E+02	2.02E+02
	p value	**6.47E−08^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**	**6.47E−08^{+}**	**5.94E−08^{+}**	**3.59E−05^{+}**	**1.24E−03^{+}**
F_{11}	Median	1.04E+01	1.05E+01	1.03E+01	1.04E+01	5.08E−13	2.17E+02	2.30E+02
	Mean	1.04E+01	1.04E+01	1.04E+01	1.05E+01	3.79E−03	2.16E+02	2.30E+02
	Std	8.03E−01	1.07E+00	9.35E−01	6.32E−01	2.07E−02	2.01E+00	2.31E−01
	p value	6.47E−08^{-}	6.47E−08^{-}	6.47E−08^{-}	6.47E−08^{-}	6.46E−08^{-}	**3.59E−05^{+}**	**4.61E−08^{+}**
F_{12}	Median	1.39E+05	4.35E+03	1.42E+03	1.32E+03	1.40E+03	2.15E+02	1.84E+07

Table 3 (continued)

F	Quality	DECC-GDG	DECC-DG2	DECC-RDG	DECC-RDG2	jDEsps	CO	eWOA
	Mean	1.39E+05	4.50E+03	1.52E+03	1.42E+03	3.19E+04	2.11E+02	1.81E+07
	Std	6.67E+03	1.13E+03	3.67E+02	2.77E+02	1.13E+05	1.70E+01	2.64E+06
	p value	**6.47E−08⁺**	**6.47E−08⁺**	**6.47E−08⁺**	**6.47E−08⁺**	**1.43E−07⁺**	**5.78E−05⁺**	**1.24E−03⁺**
F_{13}	Median	8.97E+02	1.24E+03	6.06E+02	7.21E+02	1.98E+01	1.23E+03	6.47E+11
	Mean	8.77E+02	1.53E+03	6.58E+02	8.45E+02	1.44E+02	1.41E+03	6.47E+11
	Std	1.58E+02	1.09E+03	1.94E+02	5.54E+02	2.79E+02	5.49E+02	1.48E+10
	p value	**6.47E−08⁺**	**6.47E−08⁺**	**7.40E−08⁺**	**6.47E−08⁺**	2.51E−03⁻	**3.59E−05⁺**	**2.91E−08⁺**
F_{14}	Median	4.71E+08	4.58E+08	3.42E+08	3.42E+08	2.90E+07	8.18E+07	2.16E+11
	Mean	4.68E+08	4.52E+08	3.45E+08	3.39E+08	6.52E+07	8.18E+07	2.19E+11
	Std	2.85E+07	2.65E+07	2.47E+07	2.00E+07	1.06E+08	5.15E+06	1.55E+10
	p value	**6.47E−08⁺**	**6.47E−08⁺**	**6.47E−08⁺**	**6.47E−08⁺**	**1.55E−02⁺**	**3.59E−05⁺**	**1.26E−05⁺**
F_{15}	Median	6.06E+03	6.11E+03	5.86E+03	5.86E+03	5.99E+03	1.08E+04	1.66E+04
	Mean	6.06E+03	6.10E+03	5.86E+03	5.88E+03	6.05E+03	1.07E+04	1.66E+04
	Std	1.00E+02	1.08E+02	8.44E+01	9.85E+01	1.13E+03	5.80E+02	1.50E+02
	p value	**6.47E−08⁺**	**6.47E−08⁺**	**6.47E−08⁺**	**6.47E−08⁺**	**6.46E−08⁺**	**3.59E−05⁺**	**3.37E−08⁺**
F_{16}	Median	5.49E−11	5.41E−11	2.66E−13	2.70E−13	5.13E+01	3.96E+02	4.18E+02
	Mean	4.23E−02	5.43E−11	2.73E−13	2.70E−13	4.85E+01	3.96E+02	4.18E+02
	Std	2.32E−01	5.72E−12	2.44E−14	1.25E−14	6.33E+00	6.93E−01	1.48E-01
	p value	2.84E−01⁼	2.84E−01⁼	2.83E−01⁼	2.83E−01⁼	**4.89E−08⁺**	**3.56E−05⁺**	**1.23E−03⁺**
F_{17}	Median	7.35E+04	7.39E+04	4.00E+04	4.07E+04	5.57E+03	2.65E+03	4.57E+07
	Mean	7.43E+04	7.42E+04	4.02E+04	4.05E+04	6.21E+04	2.73E+03	4.43E+07
	Std	4.34E+03	4.29E+03	2.75E+03	2.15E+03	1.71E+05	2.38E+02	4.65E+06
	p value	**6.47E−08⁺**	**6.47E−08⁺**	**6.47E−08⁺**	**6.47E−08⁺**	**8.26E−05⁺**	**1.81E−04⁺**	**1.26E−05⁺**
F_{18}	Median	1.24E+03	1.30E+03	1.16E+03	1.17E+03	1.43E+03	3.17E+03	1.42E+12
	Mean	1.24E+03	1.31E+03	1.14E+03	1.17E+03	1.85E+03	3.47E+03	1.43E+12
	Std	1.52E+02	1.59E+02	1.10E+02	1.03E+02	1.87E+03	1.02E+03	8.76E+09
	p value	**8.45E−08⁺**	**6.47E−08⁺**	**7.40E−08⁺**	**6.47E−08⁺**	**6.63E−07⁺**	**3.59E−05⁺**	**1.24E−03⁺**
F_{19}	Median	1.86E+06	1.84E+06	1.71E+06	1.73E+06	8.66E+05	6.63E+05	9.01E+07
	Mean	1.87E+06	1.85E+06	1.72E+06	1.73E+06	9.34E+05	6.77E+05	1.04E+08
	Std	1.09E+05	1.02E+05	8.45E+04	8.58E+04	3.41E+05	3.53E+04	2.41E+07
	p value	**6.47E−08⁺**	**6.47E−08⁺**	**6.47E−08⁺**	**6.47E−08⁺**	2.95E−05⁻	3.59E−05⁻	**2.47E−04⁺**
F_{20}	Median	7.04E+03	6.80E+03	3.97E+03	3.99E+03	7.09E+02	1.81E+03	1.59E+12
	Mean	1.54E+05	2.38E+04	7.34E+03	1.94E+04	7.05E+02	1.80E+03	1.58E+12
	Std	6.14E+05	4.28E+04	9.22E+03	8.04E+04	5.43E+02	1.88E+02	2.42E+10
	p value	**6.47E−08⁺**	**6.47E−08⁺**	**6.47E−08⁺**	**6.47E−08⁺**	1.66E−01⁼	**3.59E−05⁺**	**1.24E−03⁺**
w/t/l		17/1/2	17/1/2	17/1/2	17/1/2	13/1/6	18/0/2	20/0/0
Rank		10.20	9.15	6.80	7.20	5.25	7.60	15.00

The bolded *p* values mean that REELSO is significantly better than the corresponding compared methods
*The *p* value of the Friedman test is 1.93E−20

methods, namely DECC-DG, DECC-XDG, DECC-GDG, DECC-DG2, DECC-RDG, and DECC-RDG2, REELSO performs significantly better than them all on 17 problems. As compared to the three state-of-the-art other large-scale evolutionary algorithms, namely jDEsps, CO, and eWOA, REELSO performs significantly better than them on 13, 18, and 20 problems, respectively. These observations substantiate that REELSO is of more effectiveness than the 14 compared approaches in solving the 1000-*D* CEC'2010 benchmark problems.

Table 4 Performance comparison between REELSO and the 14 compared large-scale optimizers on the 1000-*D* CEC'2013 problems

F	Quality	REELSO	TPLSO	SDLSO	DLLSO	CSO	SL-PSO	DECC-DG	DECC-XDG
F_1	Median	2.40E−25	2.93E−18	4.71E−23	3.85E−22	7.91E−12	1.04E−17	3.86E+02	5.32E+02
	Mean	1.27E−24	3.08E−18	4.69E−23	3.92E−22	7.88E−12	5.80E+02	5.05E+03	5.03E+03
	Std	1.79E−24	1.26E−18	1.44E−23	9.07E−23	1.21E−12	1.83E+03	1.85E+04	1.60E+04
	p value	–	**5.37E−08^{+}**	**2.61E−09^{+}**	**2.61E−09^{+}**	**2.61E−09^{+}**	**8.00E−06^{+}**	**2.61E−09^{+}**	**2.61E−09^{+}**
F_2	Median	1.53E+03	1.18E+03	9.95E+02	1.16E+03	8.57E+03	2.09E+03	1.27E+04	1.28E+04
	Mean	1.51E+03	1.20E+03	9.94E+02	1.15E+03	8.58E+03	2.10E+03	1.26E+04	1.26E+04
	Std	1.15E+02	1.26E+02	4.05E+01	7.27E+01	1.79E+02	2.74E+01	6.99E+02	6.55E+02
	p value	–	3.84E−07^{-}	2.97E−09^{-}	3.79E−09^{-}	**2.99E−09^{+}**	**8.91E−06^{+}**	**2.99E−09^{+}**	**2.99E−09^{+}**
F_3	Median	2.16E+01	2.16E+01	2.16E+01	2.16E+01	2.16E+01	2.16E+01	2.14E+01	2.14E+01
	Mean	2.16E+01	2.16E+01	2.16E+01	2.16E+01	2.16E+01	2.16E+01	2.14E+01	2.14E+01
	Std	7.79E−03	6.36E−03	4.87E−03	7.20E−03	5.99E−03	1.58E−02	1.36E−02	1.34E−02
	p value	–	3.65E−01$^{=}$	3.08E−01$^{=}$	5.66E−01$^{=}$	9.61E−01$^{=}$	3.90E−01$^{=}$	2.99E−09^{-}	2.99E−09^{-}
F_4	Median	3.01E+08	3.23E+09	3.02E+09	6.05E+09	1.22E+10	4.53E+09	6.47E+10	8.51E+09
	Mean	2.95E+08	3.45E+09	3.02E+09	5.94E+09	1.35E+10	6.28E+09	7.50E+10	8.52E+09
	Std	1.03E+08	1.23E+09	6.28E+08	1.47E+09	3.17E+09	5.08E+09	3.90E+10	2.81E+09
	p value	–	**6.80E−08^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**1.20E−05^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**
F_5	Median	5.18E+05	6.39E+05	6.50E+05	6.54E+05	5.90E+05	1.09E+06	5.80E+06	5.40E+06
	Mean	5.27E+05	6.42E+05	6.71E+05	6.61E+05	5.97E+05	1.09E+06	5.69E+06	5.42E+06
	Std	8.18E+04	8.65E+04	1.28E+05	1.11E+05	1.04E+05	4.01E+02	5.17E+05	4.25E+05
	p value	–	**5.56E−04^{+}**	**9.15E−05^{+}**	**1.49E−04^{+}**	**7.28E−03^{+}**	**9.69E−06^{+}**	**2.99E−09^{+}**	**2.99E−09^{+}**
F_6	Median	1.06E+06	1.06E+06	1.06E+06	1.06E+06	1.06E+06	1.06E+06	1.06E+06	1.06E+06
	Mean	1.06E+06	1.06E+06	1.06E+06	1.06E+06	1.06E+06	1.06E+06	1.06E+06	1.06E+06
	Std	9.38E+02	1.17E+03	1.23E+03	9.38E+02	1.20E+03	5.24E+02	1.40E+03	1.62E+03
	p value	–	8.39E−01$^{=}$	4.76E−01$^{=}$	1.98E−01$^{=}$	1.43E−01$^{=}$	2.80E−01$^{=}$	1.57E−01$^{=}$	8.04E−01$^{=}$
F_7	Median	3.71E+05	7.50E+05	2.70E+05	1.54E+06	5.45E+06	2.07E+06	4.24E+08	1.70E+07
	Mean	4.20E+05	8.66E+05	3.47E+05	2.45E+06	5.81E+06	1.32E+07	4.78E+08	1.71E+07
	Std	2.36E+05	5.02E+05	3.05E+05	3.62E+06	3.09E+06	2.56E+07	2.38E+08	6.16E+06
	p value	–	**1.78E−03^{+}**	1.02E−01$^{=}$	**8.78E−09^{+}**	**3.01E−09^{+}**	**3.90E−05^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**
F_8	Median	1.14E+13	5.14E+13	4.58E+13	1.16E+14	2.42E+14	8.80E+13	4.31E+15	2.94E+14
	Mean	1.22E+13	5.69E+13	4.91E+13	1.28E+14	2.46E+14	1.83E+14	4.32E+15	3.21E+14
	Std	5.01E+12	2.30E+13	1.57E+13	4.70E+13	8.86E+13	2.09E+14	2.10E+15	1.86E+14
	p value	–	**1.06E−07^{+}**	**6.17E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**1.20E−05^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**
F_9	Median	3.84E+07	4.51E+07	1.07E+08	1.05E+08	5.94E+07	6.57E+07	4.86E+08	5.37E+08
	Mean	3.98E+07	4.29E+07	1.11E+08	1.17E+08	6.08E+07	6.76E+07	4.91E+08	5.35E+08
	Std	5.40E+06	8.32E+06	3.08E+07	4.22E+07	1.31E+07	5.56E+06	2.77E+07	2.26E+07
	p value	–	1.98E−01$^{=}$	**2.99E−09^{+}**	**2.99E−09^{+}**	**9.39E−08^{+}**	**1.17E−05^{+}**	**2.99E−09^{+}**	**2.99E−09^{+}**
F_{10}	Median	9.40E+07	9.42E+07	9.40E+07	9.41E+07	9.41E+07	9.40E+07	9.45E+07	9.46E+07
	Mean	9.39E+07	9.42E+07	9.40E+07	9.40E+07	9.40E+07	9.40E+07	9.45E+07	9.45E+07
	Std	2.01E+05	2.52E+05	1.89E+05	2.28E+05	2.25E+05	1.03E+05	3.23E+05	2.68E+05
	p value	–	**3.04E−03^{+}**	1.11E−01$^{=}$	2.09E−01$^{=}$	**4.04E−02^{+}**	4.09E−01$^{=}$	**3.41E−06^{+}**	**5.42E−08^{+}**
F_{11}	Median	2.37E+06	1.02E+08	9.22E+11	9.25E+11	9.25E+11	9.20E+11	1.98E+10	5.23E+08
	Mean	2.49E+06	1.17E+08	9.28E+11	9.29E+11	9.29E+11	9.20E+11	4.17E+10	5.34E+08
	Std	1.69E+06	7.52E+07	1.59E+10	9.30E+09	9.80E+09	5.79E+07	7.02E+10	1.19E+08
	p value	–	**6.80E−08^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**1.15E−05^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**

Table 4 (continued)

F	Quality	REELSO	TPLSO	SDLSO	DLLSO	CSO	SL-PSO	DECC-DG	DECC-XDG
F_{12}	Median	9.11E+02	1.96E+03	1.25E+03	1.78E+03	1.04E+03	2.00E+03	1.67E+11	2.10E+04
	Mean	9.12E+02	1.96E+03	1.27E+03	1.79E+03	1.08E+03	2.26E+03	1.65E+11	1.13E+06
	Std	6.59E+01	1.80E+02	1.15E+02	1.18E+02	7.51E+01	7.14E+02	1.61E+10	5.79E+06
	p value	–	**6.78E−08**$^{+}$	**3.01E−09**$^{+}$	**3.01E−09**$^{+}$	**3.11E−08**$^{+}$	**1.20E−05**$^{+}$	**3.01E−09**$^{+}$	**3.01E−09**$^{+}$
F_{13}	Median	3.37E+06	6.69E+07	2.55E+08	3.66E+08	7.08E+08	2.89E+09	1.99E+10	1.07E+09
	Mean	5.51E+06	1.09E+08	2.65E+08	3.80E+08	7.48E+08	3.43E+09	2.04E+10	1.11E+09
	Std	6.32E+06	1.21E+08	1.21E+08	1.59E+08	2.89E+08	3.97E+09	5.56E+09	2.48E+08
	p value	–	**6.01E−07**$^{+}$	**3.01E−09**$^{+}$	**3.01E−09**$^{+}$	**3.01E−09**$^{+}$	**1.20E−05**$^{+}$	**3.01E−09**$^{+}$	**3.01E−09**$^{+}$
F_{14}	Median	9.61E+06	3.77E+07	5.20E+07	8.99E+07	2.90E+09	1.43E+08	2.05E+10	2.62E+09
	Mean	9.89E+06	4.43E+07	6.91E+07	1.21E+08	3.67E+09	1.39E+09	2.29E+10	3.05E+09
	Std	2.43E+06	1.79E+07	4.84E+07	8.75E+07	3.38E+09	2.61E+09	1.32E+10	1.84E+09
	p value	–	**6.80E−08**$^{+}$	**3.01E−09**$^{+}$	**3.01E−09**$^{+}$	**3.01E−09**$^{+}$	**1.20E−05**$^{+}$	**3.01E−09**$^{+}$	**3.01E−09**$^{+}$
F_{15}	Median	2.73E+06	1.03E+07	1.21E+07	4.40E+06	7.60E+07	6.30E+07	9.32E+06	9.11E+06
	Mean	2.81E+06	1.06E+07	1.20E+07	4.47E+06	7.61E+07	6.70E+07	9.78E+06	9.65E+06
	Std	1.09E+06	2.18E+06	8.00E+05	4.12E+05	6.24E+06	1.09E+07	1.79E+06	1.79E+06
	p value	–	**6.80E−08**$^{+}$	**3.01E−09**$^{+}$	**1.05E−07**$^{+}$	**3.01E−09**$^{+}$	**1.20E−05**$^{+}$	**3.01E−09**$^{+}$	**3.01E−09**$^{+}$
w/t/l		–	11/3/1	10/4/1	11/3/1	13/2/0	12/3/0	13/1/1	13/1/1
Rank		**3.13**	5.33	5.93	6.00	7.53	8.40	11.07	9.20

F	Quality	DECC-GDG	DECC-DG2	DECC-RDG	DECC-RDG2	jDEsps	CO	eWOA
F_1	Median	5.04E−10	1.65E+01	1.97E−01	5.72E−01	1.43E−20	4.05E−01	1.81E+11
	Mean	5.12E−10	9.61E+04	5.17E+00	5.50E+01	2.30E−14	4.19E−01	1.80E+11
	Std	1.12E−10	5.25E+05	2.13E+01	1.96E+02	1.25E−13	4.45E−02	7.92E+09
	p value	**2.61E−09**$^{+}$	**2.61E−09**$^{+}$	**2.61E−09**$^{+}$	**2.61E−09**$^{+}$	**2.66E−07**$^{+}$	**8.19E−06**$^{+}$	**3.91E−11**$^{+}$
F_2	Median	4.56E+02	1.28E+04	1.26E+04	1.26E+04	6.12E+01	2.03E+02	4.24E+04
	Mean	4.60E+02	1.28E+04	1.27E+04	1.26E+04	1.03E+02	2.06E+02	4.23E+04
	Std	2.13E+01	5.99E+02	6.37E+02	5.42E+02	1.10E+02	1.66E+01	2.67E+02
	p value	2.99E−09^{-}	**2.99E−09**$^{+}$	**2.99E−09**$^{+}$	**2.99E−09**$^{+}$	2.98E−09^{-}	1.17E−05^{-}	**3.20E−08**$^{+}$
F_3	Median	2.14E+01	2.14E+01	2.14E+01	2.14E+01	2.00E+01	2.00E+01	2.16E+01
	Mean	2.14E+01	2.14E+01	2.14E+01	2.14E+01	2.00E+01	2.00E+01	2.16E+01
	Std	1.51E−02	1.63E−02	1.77E−02	1.75E−02	1.22E−02	3.11E−04	1.11E−02
	p value	2.99E−09^{-}	2.99E−09^{-}	2.99E−09^{-}	2.99E−09^{-}	2.99E−09^{-}	1.18E−05^{-}	2.59E−01$^{=}$
F_4	Median	2.47E+11	5.53E+10	5.08E+10	4.70E+10	1.13E+09	1.29E+09	5.93E+12
	Mean	2.50E+11	6.06E+10	4.95E+10	4.80E+10	1.89E+09	1.39E+09	6.53E+12
	Std	6.89E+10	2.31E+10	1.57E+10	1.34E+10	2.81E+09	5.88E+08	2.46E+12
	p value	**3.01E−09**$^{+}$	**3.01E−09**$^{+}$	**3.01E−09**$^{+}$	**3.01E−09**$^{+}$	**1.11E−08**$^{+}$	**1.20E−05**$^{+}$	**1.54E−07**$^{+}$
F_5	Median	7.83E+06	5.19E+06	5.12E+06	5.19E+06	3.20E+06	1.15E+07	3.63E+07
	Mean	7.83E+06	5.24E+06	5.13E+06	5.21E+06	3.22E+06	1.19E+07	3.63E+07
	Std	4.31E+05	4.21E+05	4.50E+05	4.34E+05	3.82E+05	2.07E+06	2.05E+06
	p value	**2.99E−09**$^{+}$	**2.99E−09**$^{+}$	**2.99E−09**$^{+}$	**2.99E−09**$^{+}$	**2.99E−09**$^{+}$	**1.17E−05**$^{+}$	**9.97E−08**$^{+}$
F_6	Median	1.06E+06	1.06E+06	1.06E+06	1.06E+06	1.05E+06	1.04E+06	1.06E+06
	Mean	1.06E+06	1.06E+06	1.06E+06	1.06E+06	1.05E+06	1.04E+06	1.06E+06
	Std	1.50E+03	1.50E+03	1.20E+03	1.61E+03	2.37E+03	1.18E+04	3.42E+03
	p value	**4.19E−07**$^{+}$	1.06E−01$^{=}$	**3.96E−05**$^{+}$	1.75E−01$^{=}$	3.00E−09^{-}	1.19E−05^{-}	2.97E−03^{-}
F_7	Median	3.93E+08	4.43E+07	8.89E+07	5.40E+07	7.81E+08	9.34E+04	4.72E+13

Table 4 (continued)

F	Quality	DECC-GDG	DECC-DG2	DECC-RDG	DECC-RDG2	jDEsps	CO	eWOA
	Mean	4.03E+08	4.42E+07	9.82E+07	5.69E+07	8.72E+08	8.81E+04	6.20E+13
	Std	9.76E+07	1.54E+07	4.17E+07	2.05E+07	3.47E+08	2.63E+04	3.37E+13
	p value	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	1.20E−05^{-}	**3.24E−08^{+}**
F_8	Median	9.13E+15	5.73E+15	4.96E+15	3.98E+15	9.31E+12	4.25E+13	2.83E+17
	Mean	9.51E+15	5.68E+15	4.83E+15	4.31E+15	3.34E+13	4.59E+13	2.64E+17
	Std	3.21E+15	1.99E+15	1.71E+15	1.61E+15	7.02E+13	1.53E+13	9.12E+16
	p value	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	7.44E−01$^{=}$	**1.20E−05^{+}**	**6.16E−06^{+}**
F_9	Median	5.15E+08	5.21E+08	4.87E+08	4.93E+08	2.54E+08	1.02E+09	3.04E+09
	Mean	5.09E+08	5.16E+08	4.88E+08	4.87E+08	2.80E+08	1.09E+09	3.07E+09
	Std	3.16E+07	2.97E+07	2.45E+07	3.28E+07	9.45E+07	1.41E+08	2.67E+08
	p value	**2.99E−09^{+}**	**2.99E−09^{+}**	**2.99E−09^{+}**	**2.99E−09^{+}**	**2.99E−09^{+}**	**1.17E−05^{+}**	**1.17E−05^{+}**
F_{10}	Median	9.46E+07	9.46E+07	9.46E+07	9.45E+07	9.30E+07	9.23E+07	9.34E+07
	Mean	9.46E+07	9.45E+07	9.45E+07	9.45E+07	9.29E+07	9.24E+07	9.34E+07
	Std	2.16E+05	1.95E+05	2.89E+05	2.45E+05	3.43E+05	5.09E+05	5.13E+05
	p value	**7.76E−09^{+}**	**1.75E−08^{+}**	**2.23E−07^{+}**	**3.47E−08^{+}**	2.99E−09^{-}	1.76E−05^{-}	3.21E−02^{-}
F_{11}	Median	6.50E+08	3.35E+09	5.53E+08	1.78E+09	9.33E+11	3.59E+07	1.90E+15
	Mean	6.68E+08	4.95E+09	6.67E+08	4.17E+09	9.34E+11	3.94E+07	1.82E+15
	Std	1.63E+08	4.82E+09	5.36E+08	6.33E+09	1.14E+09	2.81E+07	8.58E+14
	p value	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**1.21E−09^{+}**	**1.20E−05^{+}**	**1.20E−04^{+}**
F_{12}	Median	6.22E+03	6.24E+03	3.97E+03	4.05E+03	7.39E+02	1.77E+03	1.66E+12
	Mean	2.73E+04	7.92E+03	4.83E+03	7.62E+03	8.13E+02	1.80E+03	1.65E+12
	Std	8.84E+04	4.20E+03	3.30E+03	1.86E+04	5.16E+02	4.10E+02	2.02E+10
	p value	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	2.47E−01$^{=}$	**1.20E−05^{+}**	**3.01E−09^{+}**
F_{13}	Median	1.50E+09	1.27E+09	2.92E+09	6.78E+08	1.97E+10	7.31E+06	1.91E+15
	Mean	1.58E+09	1.30E+09	3.13E+09	6.93E+08	2.10E+10	7.77E+06	2.01E+15
	Std	4.48E+08	3.16E+08	8.61E+08	2.00E+08	5.81E+09	2.09E+06	1.20E+15
	p value	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**2.09E−02^{+}**	**1.20E−04^{+}**
F_{14}	Median	5.82E+09	6.02E+09	2.74E+09	3.05E+09	9.93E+10	2.31E+07	2.73E+15
	Mean	6.59E+09	5.72E+09	3.40E+09	3.06E+09	1.10E+11	2.33E+07	2.96E+15
	Std	4.05E+09	2.74E+09	2.23E+09	1.51E+09	5.33E+10	2.91E+06	3.98E+14
	p value	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**1.20E−05^{+}**	**7.08E−03^{+}**
F_{15}	Median	1.13E+07	1.11E+07	9.48E+06	9.32E+06	2.39E+07	1.48E+06	1.17E+11
	Mean	1.19E+07	1.14E+07	1.03E+07	9.59E+06	2.74E+07	1.49E+06	1.32E+11
	Std	3.01E+06	2.29E+06	2.18E+06	1.66E+06	1.12E+07	7.13E+04	5.14E+10
	p value	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	**3.01E−09^{+}**	1.20E−05^{-}	**1.20E−04^{+}**
w/t/l		13/0/2	13/1/1	14/0/1	13/1/1	9/2/4	9/0/6	12/1/2
Rank		10.60	10.80	9.73	8.53	6.67	4.00	13.07

The bolded *p* values mean that REELSO is significantly better than the corresponding compared methods
*The *p* value of the Friedman test is 5.74E−12

(3) Making a deep observation on the comparison results in terms of different kinds of optimization problems, we can see that (a) on the one fully separable unimodal problem, REELSO shows significant dominance to all 14 compared algorithms; (b) on the two fully separable multimodal problems, REELSO shows significantly better performance than 8 compared methods, namely CSO, SL-PSO, DECC-DG, DECC-XDG, DECC-DG2, DECC-RDG, DECC-RDG2 and eWOA and is competitive to the other 6 compared methods; (c) on the six partially separable unimodal problems, REELSO presents significant dominance to all 14 compared algorithms; (d) on the nine partially separable multimodal problems, REELSO presents significant superiority to 12 compared methods on more than five problems and is competitive to SDLSO and DLLSO; (e) on the one fully non-separable unimodal problem, REELSO shows significant dominance to 12 compared algorithms; (f) on the one fully non-separable multimodal problem,

REELSO is significantly superior to 13 compared methods and achieves competitive performance to jDEsps. As a whole, it is interesting to find that REELSO shows significantly better performance than the 14 compared methods nearly on all unimodal problems. This is because REELSO preserves faster convergence to optimal solutions than the 14 compared methods, which mainly benefits from the cooperation between the cognitive learning and the ensemble learning in the REEL learning scheme. Such a learning strategy ensures that each updated particle takes positive learning from its environment to approach optimal regions fast. Besides, on multimodal problems, REELSO also exhibits great superiority to most compared algorithms. This is mainly attributed to that REELSO is capable of better balancing exploration and exploitation to search the solution space. Specifically, the random construction of the learning environment for each non-elite particle affords high search diversity for the swarm to traverse the immense space in diverse directions. Besides, the cognitive learning and the ensemble learning in REEL provide fast convergence for the swarm to move toward optimal regions. Together, the devised REEL strategy endows the swarm with a good capability to explore the solution space with slight intensification and exploit the found optimal regions with slight diversification.

(4) To summarize, it is found that REELSO exhibits considerably equivalent performance with or even significantly better optimization performance than the 14 compared algorithms on different kinds of optimization problems. In particular, on partially separable problems, which are quite difficult to optimize but very common in real-world engineering, REELSO shows significant superiority to the 14 compared algorithms. This demonstrates that REELSO is very promising for solving complicated optimization problems.

From Tables 2 and 4, the following findings can be obtained on the 1000-*D* CEC’2013 benchmark problems:

(1) In terms of the average rank achieved from the Friedman test, it is found that on such difficult optimization problems, REELSO still obtains the smallest average rank among all 15 algorithms and its rank value is still far smaller than those of the 14 compared methods. This indicates that REELSO still performs the best over the whole 1000-*D* CEC’2013 benchmark set and its optimization performance is much superior to those of the 14 compared methods.

(2) With respect to “*w/t/l*” counted on the basis of the Wilcoxon rank sum test, on the 15 difficult problems, REELSO significantly outperforms the 14 compared methods on more than 9 problems, and only displays inferiority to them on no more than 6 problems. In particular, compared with TPLSO, SDLSO, DLLSO, CSO, and SL-PSO, REELSO exhibits significant superiority to them on 11, 10, 11, 13, and 12 problems, respectively. Competed with the six decomposition-based methods, REELSO significantly wins the competition on more than 13 problems. As for three other large-scale evolutionary algorithms, REELSO achieves significant superiority to them on 9, 9, and 12 problems, respectively. These observations verify that REELSO is much better than the 14 compared large-scale approaches in solving the difficult 1000-*D* CEC’2013 benchmark problems.

(3) In terms of different kinds of optimization problems, we can see that (a) on different types of unimodal problems, like the fully separable unimodal problems, the partially separable unimodal problems, the overlapping unimodal problems, and the fully non-separable unimodal problems, REELSO consistently obtains significantly better optimization results than nearly all 14 compared approaches; (b) on different kinds of multimodal problems, REELSO achieves significantly better performance than 8 compared methods, and attains competitive performance with 6 compared methods on the two fully separable multimodal problems; it obtains no worse optimization results than 11 compared methods on the five partially separable multimodal problems; besides, it significantly outperforms 13 compared methods on the one overlapping multimodal problem; (c) in particular, it is found that on the complicated overlapping problems and the complex fully non-separable problems, REELSO is significantly better than the 14 compared methods. Such superiority of REELSO to the 14 compared methods mainly profits from the devised REEL strategy. Such a learning strategy lets REELSO search the vast problem space with dynamic balance between exploration and exploitation. Confronted with unimodal problems, REELSO inclines the balance to exploitation of the optimal regions with slight diversification, so that the swarm could find the optimal regions fast and then intensively mines the found optimal regions subtly to get high-accuracy solutions. By contrast, in face of multimodal problems, REELSO first inclines the balance to exploration of the immense solution space with slight intensification to locate more promising areas and then inclines the balance to exploitation of the found optimal regions with slight diversification to find high-quality solutions.

(4) To sum up, REELSO exhibits considerably equivalent performance with or even significant superiority to the 14 compared algorithms on different types of

benchmark problems. In particular, confronted with partially separable problems and overlapping problems that are quite complicated but very common in real-world engineering, REELSO shows significant superiority to the 14 compared algorithms. This demonstrates that REELSO is very promising for tackling complicated optimization problems.

To further verify the effectiveness and efficiency of the devised REELSO, this paper further conducts experiments to observe the convergence behaviors of REELSO by comparing it with the 14 compared optimizers on the CEC'2010 and CEC'2013 sets. Figures 2 and 3 exhibit the convergence behaviors of REELSO and the 14 compared methods on the two high-dimensional problem sets, respectively.

From Fig. 2, close observation on the eight unimodal problems (F_1, F_4, F_7, F_9, F_{12}, F_{14}, F_{17} and F_{19}) shows that REELSO attains significantly higher solution quality along with faster convergence than the 14 compared methods on 5 problems (F_1, F_4, F_7, F_9, F_{14}). On F_{12} and F_{17}, REELSO achieves better performance in terms of solution quality and convergence speed at the early stage than the 14 compared methods. However, at the late stage, it is slightly inferior to one or two compared methods, but is still much better than the other compared methods. On the 12 multimodal problems (F_2, F_3, F_5, F_6, F_8, F_{10}, F_{11}, F_{13}, F_{15}, F_{16}, F_{18}, and F_{20}), REELSO achieves higher solution quality and faster convergence speed than at least 12 compared methods on 8 problems (F_3, F_5, F_8, F_{10}, F_{13}, F_{15}, F_{18}, and F_{20}).

From Fig. 3, similar conclusions can be drawn on the CEC'2013 problem set. Specifically, on the seven unimodal problems (F_1, F_4, F_8, F_{11}, and F_{13}–F_{15}), REELSO shows much better performance than the 14 compared methods in terms of both the solution quality and the convergence speed on five problems (F_1, F_4, F_8, F_{11}, F_{14}). On the other two problems (F_{13} and F_{15}), REELSO presents significantly better performance than 13 compared methods. On the eight multimodal problems (F_2, F_3, F_5–F_7, F_9, F_{10}, and F_{12}), REELSO shows significant superiority to at least 13 compared methods with respect to the solution quality and the convergence speed on four problems (F_5, F_9, F_7, and F_{12}).

As a whole, we find that REELSO presents significantly better performance than the 14 compared methods on the unimodal problems in the two benchmark sets, Such superiority of REELSO mainly benefits from the devised REEL strategy, which ensures that each updated particle takes positive learning from its surroundings. With the cognitive guidance of the best elite and the ensemble guidance of all elites in the randomly constructed learning environment, the updated particles are expected to move toward optimal regions fast in diverse directions. Confronted with such a kind of optimization problems, REELSO inclines the balance between exploration and exploitation to search the vast space with slight intensification. As a result, the swarm could fast locate optimal regions and then intensively exploit the found optimal areas subtly to find high-quality solutions. On the multimodal problems, REELSO also presents significant superiority to most of the 14 compared methods. This is mainly contributed by the high search diversity maintenance endowed by the devised REEL strategy. In particular, in this learning strategy, each particle in the non-elite group is provided with a positive learning environment formed by elite particles randomly chosen from the elite group in the current swarm. The random construction of the positive learning environment of each particle in the non-elite group leads to that different non-elite particles have different learning environments and thus they can take the cognitive learning and the ensemble learning from different elites. As a result, high learning diversity is maintained among particles, which likely ensures that particles in REELSO are capable of searching the multimodal space in diverse directions. Cooperated with the cognitive learning and the ensemble learning mechanisms, REELSO could explore the immense solution space with slight intensification to locate promising regions fast and exploit the found optimal areas with slight diversification to subtly find high-quality solutions.

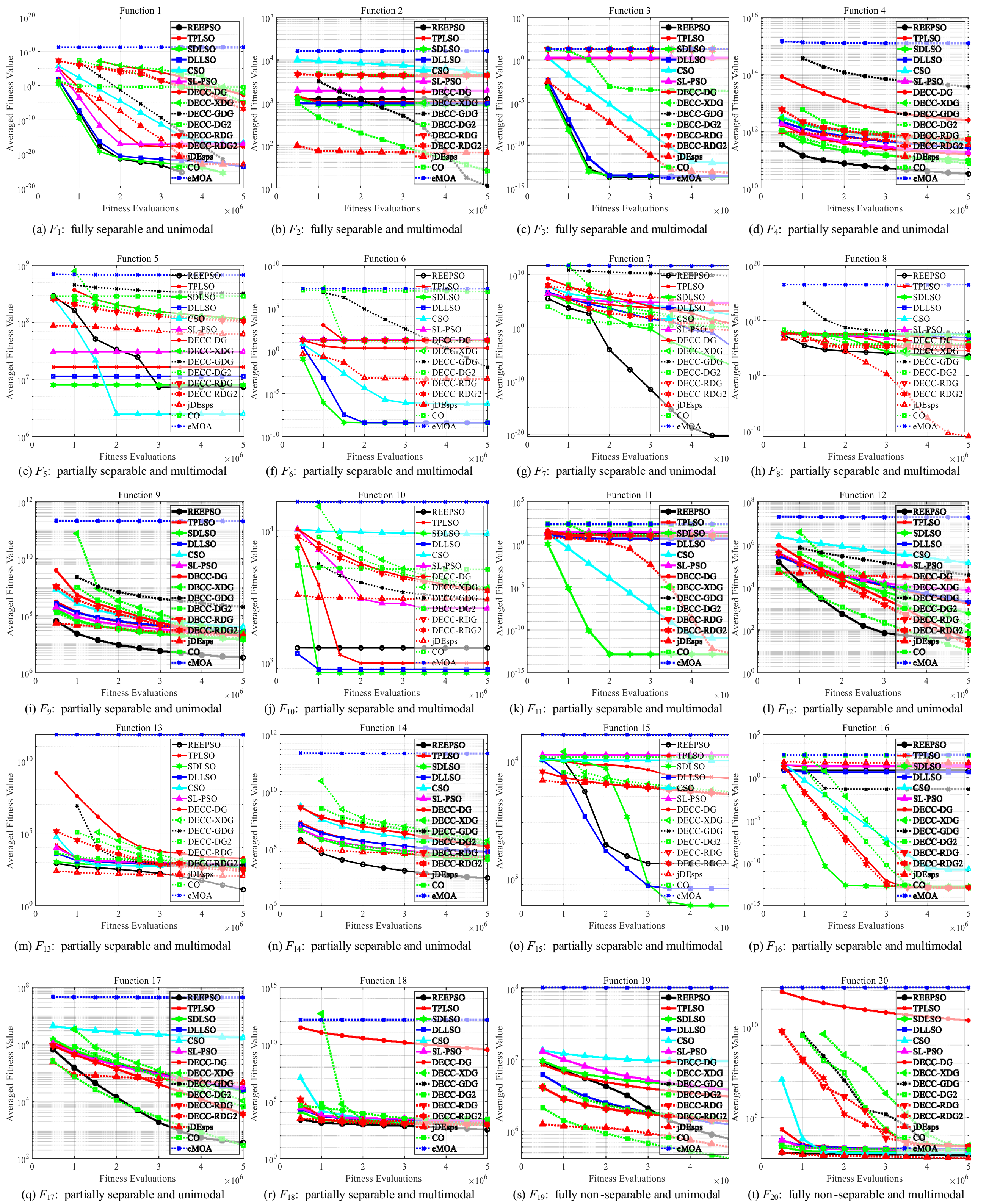
Scalability investigation

After the above extensive comparisons between REELSO and the 14 compared large-scale approaches on the two sets of 1000-*D* benchmark problems, it is interesting to further investigate the scalability of REELSO to deal with optimization problems with higher dimensionality. To this end, we carry out experiments on the CEC'2010 problem set by changing the dimension size to 2000 and compare REELSO with the 14 compared large-scale methods. In this experiment, the swarm size NP for REELSO is set as 900 and the other parameters are set the same as those used to solve the 1000-*D* problems in the last subsection. With respect to the compared large-scale optimizers, we only fine-tune their population sizes with the other parameters set according to the recommendation in the associated papers. Table 5 presents the summarized comparison results in terms of the two statistical tests, while Table 6 displays the detailed comparison results.

From Tables 5 and 6, the following findings on the 2000-*D* CEC'2010 problems can be obtained:

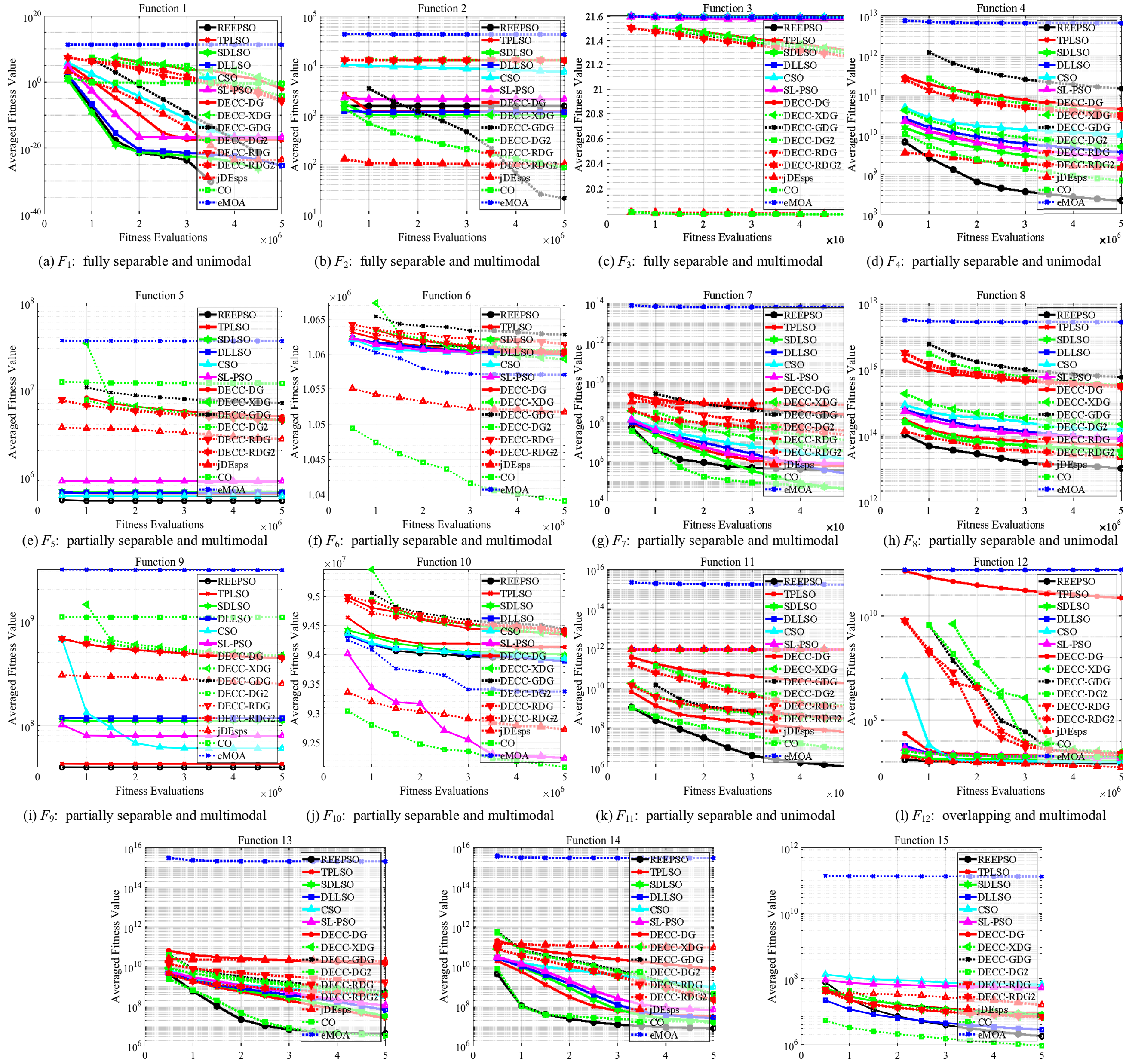
(1) Regarding the average rank achieved from the Friedman test, it is found that on such high-dimensional problems, REELSO still ranks the first among all 15 algorithms. This indicates that REELSO still performs the best on the whole 2000-*D* CEC'2010 benchmark set.

(2) With respect to "*w/t/l*" calculated from the results of the Wilcoxon rank sum test, on the 20 difficult problems,



(a) F_1: fully separable and unimodal
(b) F_2: fully separable and multimodal
(c) F_3: fully separable and multimodal
(d) F_4: partially separable and unimodal
(e) F_5: partially separable and multimodal
(f) F_6: partially separable and multimodal
(g) F_7: partially separable and unimodal
(h) F_8: partially separable and multimodal
(i) F_9: partially separable and unimodal
(j) F_{10}: partially separable and multimodal
(k) F_{11}: partially separable and multimodal
(l) F_{12}: partially separable and unimodal
(m) F_{13}: partially separable and multimodal
(n) F_{14}: partially separable and unimodal
(o) F_{15}: partially separable and multimodal
(p) F_{16}: partially separable and multimodal
(q) F_{17}: partially separable and unimodal
(r) F_{18}: partially separable and multimodal
(s) F_{19}: fully non-separable and unimodal
(t) F_{20}: fully non-separable and multimodal

Fig. 2 Convergence behavior comparison between REELSO and the 14 compared methods on the 1000-D CEC'2010 problems



(a) F_1: fully separable and unimodal (b) F_2: fully separable and multimodal (c) F_3: fully separable and multimodal (d) F_4: partially separable and unimodal

(e) F_5: partially separable and multimodal (f) F_6: partially separable and multimodal (g) F_7: partially separable and multimodal (h) F_8: partially separable and unimodal

(i) F_9: partially separable and multimodal (j) F_{10}: partially separable and multimodal (k) F_{11}: partially separable and unimodal (l) F_{12}: overlapping and multimodal

(m) F_{13}: overlapping and unimodal (n) F_{14}: overlapping and unimodal (o) F_{15}: fully non-separable and unimodal

Fig. 3 Convergence behavior comparison between REELSO and the 14 compared methods on the CEC'2013 problems

REELSO significantly outperforms the 14 compared optimizers on more than 14 problems, and only displays inferiority to them on no more than 6 problems. In particular, compared with TPLSO, SDLSO, DLLSO, CSO, and SL-PSO, REELSO exhibits significant superiority to them on 16, 14, 14, 16, and 20 problems, respectively. Competed with the six decomposition-based methods and the three other evolutionary algorithms, REELSO significantly wins the competition on more than 14 problems, and only displays inferiority to them on no more than 4 problems. These observations verify that REELSO is still more effective than the 14 compared large-scale approaches in solving the difficult 2000-*D* CEC'2010 benchmark problems.

(3) Regarding different kinds of optimization problems, we can see that (a) on the one fully separable unimodal problem, the six partially separable unimodal problems, and the one fully non-separable multimodal problem, REELSO consistently shows significant dominance to all 14 compared approaches; (b) on the two fully separable multimodal problems, REELSO achieves competitive performance with 5 compared

Table 5 Summarized statistical results between REELSO and the 14 compared methods on the 2000-*D* CEC'2010 problems

Benchmark set	Problem property	Index	REEPSO	TPLSO	SDLSO	DLLSO	CSO	SL-PSO	DECC-DG	DECC-XDG	DECC-GDG	DECC-DG2	DECC-RDG	DECC-RDG2	jDEsps	CO	eWOA
CEC2010-2000	Fully separable unimodal	*w/t/l*	–	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	0/0/1	1/0/0	1/0/0
	Fully separable multi-modal	*w/t/l*	–	1/0/1	1/0/1	1/0/1	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	2/0/0	1/0/1	1/1/0	2/0/0
	Partially separable unimodal	*w/t/l*	–	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	5/1/0	6/0/0	5/1/0
	Partially separable multi-modal	*w/t/l*	–	6/2/1	4/0/5	4/0/5	5/0/4	9/0/0	7/0/2	9/0/0	9/0/0	8/1/0	4/3/2	5/2/2	7/1/1	9/0/0	7/2/0
	Fully non-separable unimodal	*w/t/l*	–	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	0/0/1	1/0/0	1/0/0	1/0/0	0/0/1	0/0/1	0/0/1	0/1/0	1/0/0
	Fully non-separable multi-modal	*w/t/l*	–	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0
	Overall	*w/t/l*	–	16/2/2	14/0/6	14/0/6	16/0/4	20/0/0	17/0/3	20/0/0	20/0/0	19/1/0	14/3/3	15/2/3	14/2/4	18/2/0	17/3/0
	Overall	*Rank*	**2.60**	6.25	3.40	5.00	6.65	9.55	10.30	13.45	11.00	11.00	6.30	6.35	4.25	11.30	12.60

Table 6 Performance comparison between REELSO and the 14 compared methods on the 2000-D CEC'2010 problems

F	Quality	REELSO	TPLSO	SDLSO	DLLSO	CSO	SL-PSO	DECC-DG	DECC-XDG
F_1	Median	3.48E−21	6.82E−03	7.20E−21	1.73E−20	2.57E−11	1.43E+08	4.53E+07	3.70E+07
	Mean	3.47E−21	2.05E−01	7.35E−21	1.75E−20	2.63E−11	1.40E+08	6.49E+07	5.70E+07
	Std	2.04E−22	7.69E−01	5.74E−22	8.86E−22	2.53E−12	2.89E+07	7.08E+07	5.47E+07
	p value	–	**2.33E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$
F_2	Median	3.44E+03	3.00E+03	1.69E+03	1.38E+03	5.20E+03	4.13E+03	4.91E+04	4.90E+04
	Mean	3.43E+03	3.02E+03	1.68E+03	1.40E+03	5.31E+03	4.15E+03	4.91E+04	4.90E+04
	Std	9.56E+01	1.74E+02	4.82E+01	5.17E+01	8.14E+02	1.75E+02	3.23E+02	3.47E+02
	p value	–	4.93E−04^{-}	4.44E−04^{-}	4.44E−04^{-}	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$
F_3	Median	2.18E−14	3.14E+00	3.24E−14	3.95E−14	3.46E−09	6.30E+00	2.15E+01	2.15E+01
	Mean	2.18E−14	3.12E+00	3.08E−14	3.94E−14	3.46E−09	6.34E+00	2.15E+01	2.15E+01
	Std	0.00E+00	1.45E−01	1.80E−15	6.49E−16	1.83E−10	3.24E−01	7.86E−03	8.87E−03
	p value	–	**2.31E−04**$^{+}$	**1.25E−04**$^{+}$	**8.11E−08**$^{+}$	**4.37E−04**$^{+}$	**4.37E−04**$^{+}$	**4.37E−04**$^{+}$	**4.37E−04**$^{+}$
F_4	Median	4.45E+10	3.20E+11	1.95E+11	3.80E+11	6.25E+11	2.32E+12	1.00E+16	8.69E+15
	Mean	4.22E+10	3.27E+11	2.03E+11	3.62E+11	6.29E+11	2.32E+12	1.03E+16	9.57E+15
	Std	8.82E+09	1.04E+11	3.32E+10	7.02E+10	8.73E+10	3.97E+11	3.41E+15	3.22E+15
	p value	–	**2.33E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$
F_5	Median	7.97E+06	1.74E+07	5.97E+06	6.97E+06	4.98E+06	1.89E+07	1.52E+08	1.59E+08
	Mean	7.77E+06	1.73E+07	5.58E+06	6.00E+06	4.62E+06	2.02E+07	1.50E+08	1.56E+08
	Std	1.64E+06	4.60E+06	1.96E+06	1.74E+06	1.54E+06	5.90E+06	1.72E+07	2.17E+07
	p value	–	**3.39E−04**$^{+}$	1.02E−02^{-}	1.51E−02^{-}	1.72E−03^{-}	**4.44E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$
F_6	Median	1.93E+01	3.76E+00	4.00E−09	4.00E−09	2.16E−06	1.99E+01	2.13E+07	2.13E+07
	Mean	1.21E+01	3.80E+00	4.00E−09	4.00E−09	2.15E−06	1.99E+01	2.13E+07	2.13E+07
	Std	9.99E+00	3.49E−01	1.68E−24	8.24E−14	4.66E−08	2.33E−02	8.15E+04	6.90E+04
	p value	–	4.68E−01$^{=}$	7.92E−09^{-}	4.40E−04^{-}	4.45E−04^{-}	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$
F_7	Median	2.29E−19	5.14E+04	2.29E−02	1.29E+01	3.39E+04	2.22E+08	1.27E+12	1.25E+12
	Mean	5.98E−18	5.86E+04	2.93E−02	1.17E+01	3.64E+04	2.45E+08	1.71E+12	1.75E+12
	Std	1.26E−17	5.79E+04	1.71E−02	5.74E+00	1.07E+04	1.05E+08	1.06E+12	1.24E+12
	p value	–	**2.33E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$
F_8	Median	6.95E+03	3.60E+06	1.37E+05	2.61E+07	3.78E+07	4.81E+07	2.33E+17	2.30E+17
	Mean	8.81E+03	1.62E+07	4.85E+05	3.15E+07	3.78E+07	6.82E+07	2.35E+17	2.28E+17
	Std	1.07E+04	3.70E+07	1.90E+06	2.11E+07	6.02E+04	3.49E+07	3.96E+16	4.36E+16
	p value	–	**2.33E−04**$^{+}$	**4.45E−04**$^{+}$	**4.44E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$
F_9	Median	1.66E+07	1.43E+08	6.03E+07	1.05E+08	1.66E+08	1.61E+09	2.63E+08	1.77E+09
	Mean	1.70E+07	1.48E+08	6.15E+07	1.06E+08	1.67E+08	1.60E+09	2.77E+08	1.82E+09
	Std	8.52E+05	5.23E+07	3.45E+06	6.93E+06	7.57E+06	1.10E+08	4.87E+07	1.64E+08
	p value	–	**2.33E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$
F_{10}	Median	3.56E+03	2.82E+03	1.36E+03	1.16E+03	1.85E+04	4.15E+03	1.31E+04	4.95E+04
	Mean	3.52E+03	2.83E+03	1.36E+03	1.16E+03	1.85E+04	4.27E+03	1.31E+04	4.94E+04
	Std	1.53E+02	1.10E+02	4.99E+01	5.00E+01	1.61E+02	3.62E+02	3.65E+02	3.77E+02
	p value	–	3.86E−03^{-}	5.32E−03^{-}	5.32E−03^{-}	**5.32E−03**$^{+}$	**5.32E−03**$^{+}$	**5.32E−03**$^{+}$	**5.32E−03**$^{+}$
F_{11}	Median	2.02E+01	3.47E+01	3.97E−13	5.69E−13	1.21E−07	1.05E+02	1.86E+01	4.52E+02
	Mean	2.03E+01	3.53E+01	3.96E−13	5.72E−13	1.19E−07	1.06E+02	1.86E+01	4.52E+02
	Std	3.99E−01	4.57E+00	1.01E−14	2.10E−14	7.22E−09	9.55E+00	3.33E−01	1.98E−01
	p value	–	**2.33E−04**$^{+}$	4.13E−04^{-}	4.37E−04^{-}	4.45E−04^{-}	**4.45E−04**$^{+}$	4.45E−04^{-}	**4.45E−04**$^{+}$

Table 6 (continued)

F	Quality	REELSO	TPLSO	SDLSO	DLLSO	CSO	SL-PSO	DECC-DG	DECC-XDG
F_{12}	Median	1.70E+02	1.02E+05	7.53E+04	1.13E+05	4.41E+05	1.37E+06	7.56E+05	9.90E+07
	Mean	1.99E+02	1.05E+05	7.52E+04	1.12E+05	4.39E+05	1.38E+06	7.58E+05	9.93E+07
	Std	1.08E+02	6.97E+04	4.15E+03	5.91E+03	1.21E+04	7.82E+04	2.54E+04	1.13E+07
	p value	–	**2.33E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**
F_{13}	Median	6.44E+02	2.87E+03	1.23E+03	1.43E+03	1.53E+03	1.07E+07	2.02E+09	4.03E+05
	Mean	6.87E+02	4.38E+03	1.33E+03	1.48E+03	1.79E+03	1.07E+07	2.15E+09	4.02E+05
	Std	1.94E+02	5.26E+03	3.26E+02	3.13E+02	7.28E+02	2.53E+06	7.12E+08	2.58E+04
	p value	–	**2.33E−04^{+}**	**6.32E−04^{+}**	**5.31E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**
F_{14}	Median	3.84E+07	3.85E+08	1.79E+08	2.89E+08	5.19E+08	3.21E+09	6.69E+08	8.63E+11
	Mean	3.90E+07	3.99E+08	1.79E+08	2.88E+08	5.19E+08	3.35E+09	6.71E+08	8.67E+11
	Std	2.70E+06	1.41E+08	7.98E+06	9.71E+06	1.53E+07	5.69E+08	2.58E+07	2.44E+10
	p value	–	**2.33E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**
F_{15}	Median	3.65E+03	4.18E+03	2.01E+04	2.06E+04	2.02E+04	4.67E+03	1.19E+04	4.96E+04
	Mean	3.66E+03	1.19E+04	2.01E+04	2.06E+04	2.02E+04	4.86E+03	1.18E+04	4.96E+04
	Std	1.73E+02	9.15E+03	9.60E+01	7.63E+01	8.04E+01	5.25E+02	1.03E+02	3.83E+02
	p value	–	9.31E−01$^{=}$	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**
F_{16}	Median	2.23E+01	9.69E+01	5.47E−13	8.79E−01	1.66E−07	3.15E+02	4.97E−12	8.61E+02
	Mean	2.68E+01	9.28E+01	5.51E−13	8.51E−01	1.66E−07	3.17E+02	4.99E−12	8.61E+02
	Std	1.59E+01	1.40E+01	8.00E−15	9.79E−01	8.99E−09	1.32E+01	3.45E−13	3.70E−01
	p value	–	**2.32E−04^{+}**	3.64E−04^{-}	4.38E−04^{-}	4.45E−04^{-}	**4.45E−04^{+}**	4.44E−04^{-}	**4.45E−04^{+}**
F_{17}	Median	3.86E+03	4.14E+05	5.70E+05	5.86E+05	2.60E+06	2.56E+06	8.52E+04	8.29E+05
	Mean	4.46E+03	4.29E+05	5.76E+05	5.83E+05	2.62E+06	2.62E+06	8.50E+04	8.31E+05
	Std	1.85E+03	2.29E+05	1.90E+04	1.54E+04	1.04E+05	2.49E+05	3.42E+03	1.78E+04
	p value	–	**2.33E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**
F_{18}	Median	2.39E+03	9.02E+03	3.56E+03	5.04E+03	4.77E+03	2.14E+09	7.09E+10	1.64E+13
	Mean	2.29E+03	9.51E+03	3.69E+03	5.31E+03	5.22E+03	2.23E+09	7.11E+10	1.64E+13
	Std	2.51E+02	4.50E+03	7.26E+02	1.42E+03	2.34E+03	3.59E+08	7.19E+09	2.80E+11
	p value	–	**2.33E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**
F_{19}	Median	6.75E+06	1.64E+07	2.72E+07	2.77E+07	3.01E+07	1.04E+07	5.43E+06	1.10E+09
	Mean	7.99E+06	1.70E+07	2.74E+07	2.78E+07	2.98E+07	1.05E+07	5.52E+06	1.09E+09
	Std	2.47E+06	3.12E+06	1.64E+06	1.53E+06	1.70E+06	5.27E+05	3.05E+05	2.31E+08
	p value	–	**2.56E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**3.20E−02^{+}**	7.50E−04^{-}	**4.45E−04^{+}**
F_{20}	Median	1.94E+03	5.50E+03	2.43E+03	2.78E+03	2.09E+03	2.41E+09	1.69E+11	1.71E+13
	Mean	1.92E+03	5.37E+03	2.47E+03	2.79E+03	2.19E+03	2.54E+09	1.69E+11	1.71E+13
	Std	8.77E+01	8.37E+02	1.37E+02	2.45E+02	2.51E+02	5.03E+08	1.49E+10	4.16E+11
	p value	–	**2.33E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**1.72E−03^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**
w/t/l		–	16/2/2	14/0/6	14/0/6	16/0/4	20/0/0	17/0/3	20/0/0
Rank		**2.60**	6.25	3.40	5.00	6.65	9.55	10.30	13.45

F	Quality	DECC-GDG	DECC-DG2	DECC-RDG	DECC-RDG2	jDEsps	CO	eWOA
F_{1}	Median	9.31E−06	2.81E+06	9.25E+04	7.15E+04	3.70E−22	7.91E+04	3.74E+11
	Mean	9.40E−06	5.24E+06	6.74E+06	3.80E+05	7.90E−22	6.88E+04	3.73E+11
	Std	1.53E−06	8.12E+06	3.24E+07	5.93E+05	1.21E−21	3.87E+04	8.80E+09
	p value	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	**4.45E−04^{+}**	1.05E−03^{-}	**7.94E−03^{+}**	**9.99E−04^{+}**
F_{2}	Median	3.68E+03	1.21E+04	1.20E+04	1.20E+04	4.75E+02	1.16E+04	3.30E+04

Table 6 (continued)

F	Quality	DECC-GDG	DECC-DG2	DECC-RDG	DECC-RDG2	jDEsps	CO	eWOA
	Mean	3.68E+03	1.22E+04	1.20E+04	1.20E+04	7.96E+02	1.16E+04	3.29E+04
	Std	5.04E+01	2.42E+02	2.68E+02	3.97E+02	7.39E+02	6.20E+03	5.85E+01
	p value	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	4.45E−04^{-}	5.19E−02$^{=}$	**7.94E−03**$^{+}$
F_3	Median	2.15E+01	2.15E+01	1.89E+01	1.89E+01	1.71E−13	6.87E+01	2.10E+01
	Mean	2.15E+01	2.15E+01	1.89E+01	1.89E+01	2.06E−13	6.87E+01	2.10E+01
	Std	8.29E−03	9.66E−03	6.94E−02	7.62E−02	2.01E−13	3.66E+01	1.11E-02
	p value	**4.37E−04**$^{+}$	**4.37E−04**$^{+}$	**4.36E−04**$^{+}$	**4.37E−04**$^{+}$	**4.35E−04**$^{+}$	**4.33E−03**$^{+}$	**1.03E−03**$^{+}$
F_4	Median	9.51E+15	9.34E+15	3.57E+11	3.69E+11	1.27E+11	3.03E+12	1.20E+15
	Mean	9.87E+15	9.46E+15	3.93E+11	4.00E+11	1.72E+11	3.21E+12	1.48E+15
	Std	2.40E+15	2.80E+15	1.93E+11	1.29E+11	1.32E+11	1.27E+12	6.66E+14
	p value	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**6.32E−04**$^{+}$	**7.94E−03**$^{+}$	**7.94E−03**$^{+}$
F_5	Median	3.74E+08	1.11E+08	9.43E+07	9.59E+07	6.20E+07	9.25E+08	5.48E+08
	Mean	3.72E+08	1.11E+08	9.47E+07	9.89E+07	6.34E+07	9.29E+08	5.50E+08
	Std	1.74E+07	2.13E+07	1.82E+07	1.94E+07	8.43E+06	4.29E+08	3.75E+07
	p value	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**7.94E−03**$^{+}$	**2.33E−04**$^{+}$
F_6	Median	2.16E+06	1.82E+01	1.90E+01	1.90E+01	4.85E−09	4.34E+07	2.04E+07
	Mean	2.13E+06	1.82E+01	1.90E+01	1.91E+01	6.31E−09	4.16E+07	2.05E+07
	Std	2.37E+05	2.00E−01	7.46E−02	8.68E−02	3.52E−09	2.40E+07	7.63E+04
	p value	**4.45E−04**$^{+}$	4.94E−01$^{=}$	4.94E−01$^{=}$	4.94E−01$^{=}$	4.44E−04^{-}	**7.94E−03**$^{+}$	**7.94E−03**$^{+}$
F_7	Median	1.80E+12	1.30E+12	1.21E+04	6.14E+04	9.34E+02	3.50E+04	1.98E+11
	Mean	2.10E+12	1.60E+12	1.29E+04	5.89E+04	2.74E+03	3.51E+04	2.21E+11
	Std	1.38E+12	8.53E+11	6.56E+03	1.13E+04	4.07E+03	1.48E+04	7.10E+10
	p value	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**7.94E−03**$^{+}$	**9.99E−04**$^{+}$
F_8	Median	2.15E+17	2.10E+17	1.55E+04	5.58E+04	3.13E−15	7.39E+07	3.67E+16
	Mean	2.18E+17	2.15E+17	1.60E+04	7.22E+05	4.60E+06	6.75E+07	3.67E+16
	Std	3.64E+16	3.96E+16	6.73E+03	1.51E+06	1.74E+07	2.93E+07	1.63E+15
	p value	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	6.95E−02$^{=}$	**4.45E−04**$^{+}$	**6.72E−03**$^{+}$	**4.33E−03**$^{+}$	**1.59E−02**$^{+}$
F_9	Median	1.03E+09	4.51E+08	1.71E+08	1.61E+08	2.07E+07	1.32E+09	3.75E+11
	Mean	1.03E+09	4.50E+08	1.84E+08	1.80E+08	4.46E+07	1.31E+09	3.73E+11
	Std	3.98E+07	4.89E+07	5.30E+07	6.06E+07	7.64E+07	7.27E+08	7.89E+09
	p value	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	1.95E−01$^{=}$	**7.94E−03**$^{+}$	**1.55E−03**$^{+}$
F_{10}	Median	9.12E+03	1.72E+04	1.08E+04	1.08E+04	8.05E+03	3.69E+04	3.34E+04
	Mean	9.11E+03	1.72E+04	1.07E+04	1.08E+04	8.32E+03	3.68E+04	3.34E+04
	Std	8.72E+01	2.66E+02	3.03E+02	3.02E+02	1.02E+03	1.93E+04	2.13E+02
	p value	**5.32E−03**$^{+}$	**5.32E−03**$^{+}$	**5.32E−03**$^{+}$	**5.32E−03**$^{+}$	**5.32E−03**$^{+}$	**3.57E−02**$^{+}$	5.71E-02$^{=}$
F_{11}	Median	4.52E+02	4.52E+02	1.74E+01	1.73E+01	2.90E+01	1.25E+03	4.40E+02
	Mean	4.52E+02	4.52E+02	1.73E+01	1.73E+01	3.59E+01	1.25E+03	4.40E+02
	Std	2.58E−01	2.28E−01	4.84E−01	3.39E−01	9.79E+00	6.57E+02	3.73E-01
	p value	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	4.45E−04^{-}	4.45E−04^{-}	**1.19E−04**$^{+}$	**7.94E−03**$^{+}$	**9.99E−04**$^{+}$
F_{12}	Median	1.02E+08	1.01E+08	8.77E+04	8.69E+04	4.01E+03	8.30E+05	5.48E+07
	Mean	9.98E+07	9.92E+07	9.02E+04	8.74E+04	4.98E+04	8.36E+05	5.48E+07
	Std	1.21E+07	1.02E+07	1.91E+04	1.42E+04	1.20E+05	4.29E+05	6.15E+06
	p value	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**4.45E−04**$^{+}$	**1.59E−02**$^{+}$	9.52E-02$^{=}$
F_{13}	Median	2.29E+03	1.66E+05	8.61E+04	8.43E+04	8.83E+02	4.55E+04	1.49E+12
	Mean	2.31E+03	1.66E+05	8.60E+04	8.42E+04	8.51E+02	5.34E+04	1.49E+12
	Std	3.92E+02	2.05E+04	1.32E+04	1.34E+04	4.04E+02	3.53E+04	7.75E+09

Table 6 (continued)

F	Quality	DECC-GDG	DECC-DG2	DECC-RDG	DECC-RDG2	jDEsps	CO	eWOA
	p value	**4.45E−04⁺**	**4.45E−04⁺**	**4.45E−04⁺**	**4.45E−04⁺**	4.37E−01⁼	**7.94E−03⁺**	**2.33E−04⁺**
F_{14}	Median	2.31E+09	2.32E+09	6.45E+08	6.59E+08	6.33E+07	4.02E+09	4.25E+11
	Mean	2.30E+09	2.32E+09	6.47E+08	6.55E+08	1.21E+08	4.07E+09	4.23E+11
	Std	6.97E+07	1.05E+08	2.84E+07	3.09E+07	1.60E+08	2.08E+09	5.78E+09
	p value	**4.45E−04⁺**	**4.45E−04⁺**	**4.45E−04⁺**	**4.45E−04⁺**	**4.45E−04⁺**	**7.94E−03⁺**	**1.59E−02⁺**
F_{15}	Median	4.98E+04	4.97E+04	1.18E+04	1.18E+04	1.48E+04	7.00E+04	3.36E+04
	Mean	4.98E+04	4.97E+04	1.18E+04	1.18E+04	1.48E+04	6.97E+04	3.36E+04
	Std	3.87E+02	4.32E+02	1.41E+02	1.23E+02	1.66E+01	3.62E+04	1.49E+02
	p value	**4.45E−04⁺**	**4.45E−04⁺**	**4.45E−04⁺**	**4.45E−04⁺**	**3.02E−05⁺**	**7.94E−03⁺**	**9.99E−04⁺**
F_{16}	Median	8.61E+02	8.61E+02	9.88E−13	9.68E−13	2.38E+02	2.38E+03	8.38E+02
	Mean	8.61E+02	8.61E+02	9.89E−13	9.75E−13	2.48E+02	2.38E+03	8.38E+02
	Std	3.13E−01	3.12E−01	5.94E−14	4.13E−14	9.00E+01	1.25E+03	3.96E-01
	p value	**4.45E−04⁺**	**4.45E−04⁺**	4.35E−04⁻	4.42E−04⁻	**3.55E−04⁺**	**7.94E−03⁺**	**3.57E−02⁺**
F_{17}	Median	2.76E+05	2.73E+05	8.36E+04	8.35E+04	2.09E+04	2.57E+06	9.38E+07
	Mean	2.76E+05	2.73E+05	8.38E+04	8.36E+04	1.32E+05	2.58E+06	9.43E+07
	Std	1.09E+04	9.11E+03	3.10E+03	3.71E+03	3.27E+05	1.39E+06	4.76E+06
	p value	**4.45E−04⁺**	**4.45E−04⁺**	**4.45E−04⁺**	**4.45E−04⁺**	**6.72E−03⁺**	**1.59E−02⁺**	**1.59E−02⁺**
F_{18}	Median	4.67E+03	4.78E+03	2.35E+03	2.34E+03	3.38E+03	1.88E+05	3.03E+12
	Mean	4.66E+03	4.72E+03	2.39E+03	2.37E+03	5.72E+03	1.78E+05	3.03E+12
	Std	3.73E+02	4.41E+02	1.95E+02	1.81E+02	1.14E+04	8.35E+04	1.86E+10
	p value	**4.45E−04⁺**	**4.45E−04⁺**	8.69E−01⁼	9.81E−01⁼	**7.73E−03⁺**	**7.94E−03⁺**	9.52E-02⁼
F_{19}	Median	1.02E+09	9.84E+08	5.30E+06	5.37E+06	3.40E+06	1.27E+07	3.75E+08
	Mean	1.01E+09	9.87E+08	5.32E+06	5.38E+06	3.44E+06	1.27E+07	3.68E+08
	Std	2.12E+08	2.43E+08	2.91E+05	2.71E+05	8.98E+05	6.12E+06	4.48E+07
	p value	**4.45E−04⁺**	**4.45E−04⁺**	6.32E−04⁻	4.45E−04⁻	6.32E−04⁻	3.81E−01⁼	**3.57E−02⁺**
F_{20}	Median	1.71E+13	1.72E+13	3.35E+07	8.17E+06	2.64E+03	3.18E+04	3.17E+12
	Mean	1.71E+13	1.71E+13	2.32E+08	7.52E+07	2.55E+03	3.13E+04	3.17E+12
	Std	3.77E+11	3.65E+11	5.97E+08	1.56E+08	6.22E+02	1.60E+04	2.57E+10
	p value	**4.45E−04⁺**	**4.45E−04⁺**	**4.45E−04⁺**	**4.45E−04⁺**	**1.73E−02⁺**	**4.33E−03⁺**	**7.94E−03⁺**
w/t/l		20/0/0	19/1/0	14/3/3	15/2/3	14/2/4	18/2/0	17/3/0
Rank		11.00	11.00	6.30	6.35	4.25	11.30	12.60

The bolded *p* values mean that REELSO is significantly better than the corresponding compared methods
*The *p* value of the Friedman test is 3.70E−29

methods (TPLSO, SDLSO, DLLSO, jDEsps and CO), but is significantly superior to the other nine compared optimizers; (c) on the nine partially separable multimodal problems, REELSO significantly outperforms 11 compared algorithms on more than five problems and achieves highly competitive performance with SDLSO and DLLSO.

(4) Overall, it is found that confronted with such high-dimensional problems, REELSO still exhibits considerably equivalent performance with or even significantly better optimization results than the 14 compared algorithms on different kinds of optimization problems. In particular, on partially separable problems that are quite common in real-world engineering, REELSO shows significant superiority to the 14 compared algorithms. This further demonstrates REELSO is very promising for solving complex optimization problems.

Based on the above experiments, it is found that REELSO preserves a good scalability to solve large-scale problems. Such a good property of REELSO also profits from the devised REEL scheme, which provides powerful strength for REELSO to compromise the diversity and the convergence of the swarm well to search high-dimensional space.

Conclusion

Taking inspiration from the human observational learning theory proposed by Bandura [60], this paper has proposed a random elite ensemble learning swarm optimizer (REELSO) to cope with high-dimensional optimization problems. In this approach, the swarm is first partitioned into the elite group and the non-elite group according to the fitness of particles. Then, for each particle in the non-elite group, several elites are randomly selected from the elite group to form a random elite neighbor region, which acts as the learning environment of the non-elite particle. Then, the non-elite particle takes positive learning by watching and imitating the behaviors of its surroundings by cognitively learning from the best elite and then collectively learning from all elites in the learning environment. With this mechanism, each particle in the non-elite group is expected to compromise exploration and exploitation well to seek the global optimum in the large-scale space. To further help the optimizer make a good compromise between diversity and convergence, this paper additionally designed an adaptive swarm partition scheme by dynamically adjusting the size of the elite group. With this strategy, REELSO gradually changes from exploring the solution space to exploiting the found optimal zones without seriously sacrificing the search diversity.

Extensive experiments have been carried out on the widely used CEC'2010 and CEC'2013 high-dimensional benchmark sets to substantiate the effectiveness and efficiency of REELSO. In competition with 14 state-of-the-art optimizers designed for high-dimensional optimization, REELSO exhibits significant dominance to them. Additionally, experiments on higher-dimensional problems have also demonstrated that REELSO preserves a good scalability to deal with large-scale optimization. Particularly, it is experimentally found that REELSO is very promising for complicated high-dimensional problems, like partially separable problems and overlapping problems as demonstrated by the extensive experiments.

In the future, we will focus on advancing REELSO in two directions. One is to develop adaptive parameter adjustment strategies to reduce the effort in fine-tuning parameters by utilizing the evolutionary information of the swarm and particles. The other is to employ REELSO to tackle real-world optimization problems in engineering and academics, like constrained optimization problems [25–28], expensive optimization problems [13], and multi-objective optimization [3, 4].

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s40747-023-00993-w.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant 62006124 and U20B2061, 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.

Data availability Data will be made available on request.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Ethical approval All authors have checked the manuscript and approved to submit to *Complex and Intelligent Systems*. This paper has not been published elsewhere nor has it been submitted for publication elsewhere.

References

1. Wang S, Liu J, Jin Y (2020) Surrogate-assisted robust optimization of large-scale networks based on graph embedding. IEEE Trans Evol Comput 24(4):735–749
2. Chen WN, Tan DZ, Yang Q, Gu T, Zhang J (2020) Ant colony optimization for the control of pollutant spreading on social networks. IEEE Trans Cybern 50(9):4053–4065
3. Du W, Zhong W, Tang Y, Du WL, Jin Y (2019) High-dimensional robust multi-objective optimization for order scheduling: a decision variable classification approach. IEEE Trans Ind Inform 15(1):293–304
4. Ma L, Li N, Guo Y, Wang X, Yang S, Huang M, Zhang H (2022) Learning to optimize: reference vector reinforcement learning adaption to constrained many-objective optimization of industrial copper burdening system. IEEE Trans Cybern 52(12):12698–12711
5. Zhou Z, Yu H, Mumtaz S, Al-Rubaye S, Tsourdos A, Hu RQ (2020) Power control optimization for large-scale multi-antenna systems. IEEE Trans Wirel Commun 19(11):7339–7352
6. Gao XD, Cao WJ, Yang Q, Wang HL, Wang XL, Jin G, Zhang J (2022) Parameter optimization of control system design for uncertain wireless power transfer systems using modified genetic algorithm. CAAI Trans Intell Technol 7(4):582–593
7. Wang Y, Ru ZY, Wang K, Huang PQ (2020) Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing. IEEE Trans Cybern 50(9):3984–3997
8. Lu Z, Liang S, Yang Q, Du B (2022) Evolving block-based convolutional neural network for hyperspectral image classification. IEEE Trans Geosci Remote Sens 60:1–21
9. Zhang X, Zhou K, Pan H, Zhang L, Zeng X, Jin Y (2020) A network reduction-based multiobjective evolutionary algorithm for community detection in large-scale complex networks. IEEE Trans Cybern 50(2):703–716
10. Mahdavi S, Shiri ME, Rahnamayan S (2015) Metaheuristics in large-scale global continues optimization: a survey. Inf Sci 295:407–428

11. Jian JR, Zhan ZH, Zhang J (2020) Large-scale evolutionary optimization: a survey and experimental comparative study. Int J Mach Learn Cybern 11(3):729–745
12. LaTorre A, Muelas S, Peña JM (2015) A comprehensive comparison of large scale global optimizers. Inf Sci 316:517–549
13. Regis RG (2014) Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions. IEEE Trans Evol Comput 18(3):326–347
14. Omidvar MN, Li XD, Tang K (2015) Designing benchmark problems for large-scale continuous optimization. Inf Sci 316:419–436
15. Yang Q, Chen WN, Yu Z, Gu T, Li Y, Zhang H, Zhang J (2017) Adaptive multimodal continuous ant colony optimization. IEEE Trans Evol Comput 21(2):191–205
16. Yang Q, Chen WN, Li Y, Chen CLP, Xu XM, Zhang J (2017) Multimodal estimation of distribution algorithms. IEEE Trans Cybern 47(3):636–650
17. Yao J, Liu X, Zhu X, Guan H (2015) Control of large-scale systems through dimension reduction. IEEE Trans Serv Comput 8(4):563–575
18. Yang Q, Hua LT, Gao XD, Xu DD, Lu ZY, Jeon SW, Zhang J (2022) Stochastic cognitive dominance leading particle swarm optimization for multimodal problems. Mathematics 10(5):761
19. Ma L, Cheng S, Shi Y (2021) Enhancing learning efficiency of brain storm optimization via orthogonal learning design. IEEE Trans Syst Man Cybern Syst 51(11):6723–6742
20. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the international symposium on micro machine and human science, pp 39–43
21. Eberhart R, Shi YH, Kennedy JL (2001) Swarm intelligence. Morgan Kaufmann
22. Cao Y, Zhang H, Li W, Zhou M, Zhang Y, Chaovalitwongse WA (2019) Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions. IEEE Trans Evol Comput 23(4):718–731
23. Yue C, Qu B, Liang J (2018) A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems. IEEE Trans Evol Comput 22(5):805–817
24. Liu W, Wang Z, Yuan Y, Zeng N, Hone K, Liu X (2021) A novel sigmoid-function-based adaptive weighted particle swarm optimizer. IEEE Trans Cybern 51(2):1085–1093
25. Aguirre AH, Zavala AM, Diharce EV, Rionda SB (2007) COPSO: constrained optimization via PSO algorithm. Center for Research in Mathematics Technical Report No. I-07-04/22-02-2007:77
26. Rosso MM, Cucuzza R, Di Trapani F, Marano GC (2021) Nonpenalty machine learning constraint handling using PSO-SVM for structural optimization. Adv Civ Eng Mater 2021:6617750
27. Parsopoulos KE, Vrahatis MN (2002) Particle swarm optimization method for constrained optimization problems. Intell Technol Theory Appl New Trends Intell Technol 76:214–220
28. Liang JJ, Suganthan PN (2006) Dynamic multi-swarm particle swarm optimizer with a novel constraint-handling mechanism. In: IEEE international conference on evolutionary computation, pp 9–16
29. Rosso MM, Cucuzza R, Aloisio A, Marano GC (2022) Enhanced multi-strategy particle swarm optimization for constrained problems with an evolutionary-strategies-based unfeasible local search operator. Appl Sci 12(5):2285
30. De Campos JA, Pozo ATR, Duarte EP (2019) Parallel multi-swarm PSO strategies for solving many objective optimization problems. J Parallel Distrib Comput 126:13–33
31. Wei FF, Chen WN, Yang Q, Deng J, Luo XN, Jin H, Zhang J (2021) A classifier-assisted level-based learning swarm optimizer for expensive optimization. IEEE Trans Evol Comput 25(2):219–233
32. Zhang J, Zhu X, Wang Y, Zhou M (2019) Dual-environmental particle swarm optimizer in noisy and noise-free environments. IEEE Trans Cybern 49(6):2011–2021
33. Cao W, Liu K, Wu M, Xu S, Zhao J (2019) An improved current control strategy based on particle swarm optimization and steady-state error correction for SAPF. IEEE Trans Ind Appl 55(4):4268–4274
34. Slowik A, Kwasnicka H (2018) Nature inspired methods and their industry applications—swarm intelligence algorithms. IEEE Trans Ind Inform 14(3):1004–1015
35. Quaranta G, Marano GC, Greco R, Monti G (2014) Parametric identification of seismic isolators using differential evolution and particle swarm optimization. Appl Soft Comput 22:458–464
36. Quaranta G, Monti G, Marano GC (2010) Parameters identification of Van der Pol-Duffing oscillators via particle swarm optimization and differential evolution. Mech Syst Signal Process 24(7):2076–2095
37. Zhan ZH, Zhang J (2010) Self-Adaptive differential evolution based on PSO learning strategy. In: Proceedings of conference genetics evolutionary computation, pp 39–46
38. Wang H, Liang MN, Sun CL, Zhang GC, Xie LP (2021) Multiple-strategy learning particle swarm optimization for large-scale optimization problems. Complex Intell Syst 7:1–16
39. Song XF, Zhang Y, Guo YN, Sun XY, Wang YL (2020) Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data. IEEE Trans Evol Comput 24(5):882–895
40. Tang K, Li XD, Suganthan PN, Yang Z, Weise T (2010) Benchmark functions for the CEC 2010 special session and competition on large-scale global optimization. Nat Inspired ComputAppl Lab, Univ Sci Technol China Anhui China Tech Rep
41. Ma X, Li X, Zhang Q, Tang K, Liang Z, Xie W, Zhu Z (2019) A survey on cooperative co-evolutionary algorithms. IEEE Trans Evol Comput 23(3):421–441
42. Yang Q, Zhang KX, Gao XD, Xu DD, Lu ZY, Jeon SW, Zhang J (2022) A dimension group-based comprehensive elite learning swarm optimizer for large-scale optimization. Mathematics 10(7):1072
43. Chen WN, Zhang J, Lin Y, Chen E (2013) Particle swarm optimization with an aging leader and challengers. IEEE Trans Evol Comput 17(2):241–258
44. Potter MA (1997) The design and analysis of a computational model of cooperative coevolution. Dissertation, George Mason University
45. Bergh FVD, Engelbrecht AP (2004) A cooperative approach to particle swarm optimization. IEEE Trans Evol Comput 8(3):225–239
46. Xd LI, Yao X (2012) Cooperatively coevolving particle swarms for large scale optimization. IEEE Trans Evol Comput 16(2):210–224
47. Cheng R, Jin YC (2015) A competitive swarm optimizer for large scale optimization. IEEE Trans Cybern 45(2):191–204
48. Cheng R, Jin YC (2015) A social learning particle swarm optimization algorithm for scalable optimization. Inf Sci 291:43–60
49. Yang Q, Chen WN, Deng JD, Li Y, Gu T, Zhang J (2018) A level-based learning swarm optimizer for large-scale optimization. IEEE Trans Evol Comput 22(4):578–594
50. Omidvar MN, Li XD, Mei Y, Yao X (2014) Cooperative co-evolution with differential grouping for large scale optimization. IEEE Trans Evol Comput 18(3):378–393
51. Sun Y, Kirley M, Halgamuge SK (2018) A recursive decomposition method for large scale continuous optimization. IEEE Trans Evol Comput 22(5):647–661
52. Yang Q, Chen WN, Zhang J (2018) Evolution consistency based decomposition for cooperative coevolution. IEEE Access 6:51084–51097
53. Xie HY, Yang Q, XM Hu, WN Chen (2016) Cross-generation elites guided particle swarm optimization for large scale optimization. In: IEEE symposium series on computational intelligence, pp 1–8
54. Yang Q, Chen WN, Gu T, Zhang H, Yuan H, Kwong S, Zhang J (2020) A distributed swarm optimizer with adaptive

communication for large-scale optimization. IEEE Trans Cybern 50(7):3393–3408
55. Lan R, Zhu X, Lu H, Liu Z, Luo X (2020) A two-phase learning-based swarm optimizer for large-scale optimization. IEEE Trans Cybern 51(12):6284–6293
56. Yang Q, Chen WN, Gu T, Jin Y, Mao W, Zhang J (2022) An adaptive stochastic dominant learning swarm optimizer for high-dimensional optimization. IEEE Trans Cybern 52(3):1960–1976
57. Li DY, Guo WA, Lerch A, Li YM, Wang L, Wu QD (2021) An adaptive particle swarm optimizer with decoupled exploration and exploitation for large scale optimization. Swarm Evol Comput 60:100789
58. Bandura A, McClelland DC (1977) Social learning theory. Prentice Hall, Englewood Cliffs
59. Bandura A, Walters RH (1963) Social learning and personality development. Holt, Rinehart, & Winston
60. Bandura A (1986) Social foundations of thought and action. Englewood Cliffs
61. Li XD, Tang K, Omidvar MN, Yang ZY, Qin K (2013) Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization. EvolComput Mach Learn Group, RMIT Univ, Melbourne, VIC, Australia, tech rep
62. Ratnaweera A, Halgamuge SK, Watson HC (2004) Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. IEEE Trans Evol Comput 8(3):240–255
63. Seo JH, Im CH, Heo CG, Kim JK, Jung HK, Lee CG (2006) Multimodal function optimization based on particle swarm optimization. IEEE Trans Magn 42(4):1095–1098
64. Ren Z, Zhang A, Wen C, Feng Z (2014) A scatter learning particle swarm optimization algorithm for multimodal problems. IEEE Trans Cybern 44(7):1127–1140
65. Liang JJ, Qin AK, Suganthan PM, Baskar S (2004) Particle swarm optimization algorithms with novel learning strategies. IEEE Int Conf Syst Man Cybern 4:3659–3664
66. Yang Q, Jing YF, Gao XD, Xu DD, Lu ZY, Jeon SW, Zhang J (2022) Predominant cognitive learning particle swarm optimization for global numerical optimization. Mathematics 10(10):1620
67. Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans Evol Comput 10(3):281–295
68. Zhan ZH, Zhang J, Li Y, Shi Y (2011) Orthogonal learning particle swarm optimization. IEEE Trans Evol Comput 15(6):832–847
69. Yang Q, Guo X, Gao XD, Xu DD, Lu ZY (2022) Differential elite learning particle swarm optimization for global numerical optimization. Mathematics 10(8):1261
70. Akbari R, Ziarati K (2011) A rank based particle swarm optimization algorithm with dynamic adaptation. J Comput App Math 235(8):2694–2714
71. Yang Q, Bian YW, Gao XD, Xu DD, Lu ZY, Jeon SW, Zhang J (2022) Stochastic triad topology based particle swarm optimization for global numerical optimization. Mathematics 10(7):1032
72. Caraffini F, Neri F, Iacca G (2017) Large scale problems in practice: the effect of dimensionality on the interaction among variables. In: European conference on applied evolutionary computation, pp 636–652
73. Zhang Y, Chiang H (2017) A novel consensus-based particle swarm optimization-assisted trust-tech methodology for large-scale global optimization. IEEE Trans Cybern 47(9):2717–2729
74. Sun Y, Kirley M, Halgamuge SK (2015) Extended differential grouping for large scale global optimization with direct and indirect variable interactions. In: Proceedings of conference genetic and evolutionary computation, pp 313–320
75. Mei Y, Omidvar MN, Li XD, Yao X (2016) A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization. ACM Trans Math Softw 42(2):1–24
76. Omidvar MN, Yang M, Mei Y, Li X, Yao X (2017) DG2: a faster and more accurate differential grouping for large-scale black-box optimization. IEEE Trans Evol Comput 21(6):929–942
77. Sun Y, Omidvar MN, Kirley M, Li XD (2018) Adaptive threshold parameter estimation with recursive differential grouping for problem decomposition. In: Proceedings of genetic and evolutionary computation conference, pp 889–896
78. Yang M, Zhou A, Li C, Yao X (2021) An efficient recursive differential grouping for large-scale continuous problems. IEEE Trans Evol Comput 25(1):159–171
79. Molina D, Lozano M, Herrera F (2010) MA-SW-chains: memetic algorithm based on local search chains for large scale continuous global optimization. In: Proceedings of IEEE congress on evolutionary computation, pp 1–8
80. Zhao SZ, Liang JJ, Suganthan PN, Tasgetiren MF (2008) Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization. In: Proceedings of IEEE congress on evolutionary computation, pp 3845–3852
81. Cheng R, Sun CL, Jin YC (2013) A multi-swarm evolutionary framework based on a feedback mechanism. In: IEEE congress on evolutionary computation, pp 718–724
82. Ali AF, Tawhid MA (2017) A hybrid particle swarm optimization and genetic algorithm with population partitioning for large scale optimization problems. Ain Shams Eng J 8(2):191–206
83. Yang Q, Chen WN, Gu T, Zhang H, Deng JD, Li Y, Zhang J (2017) Segment-based predominant learning swarm optimizer for large-scale optimization. IEEE Trans Cybern 47(9):2896–2910
84. Deng HB, Peng LZ, Zhang HB, Yang B, Chen ZX (2019) Ranking-based biased learning swarm optimizer for large-scale optimization. Inf Sci 493:120–137
85. Wang ZJ, Zhan ZH, Kwong S, Jin H, Zhang J (2021) Adaptive granularity learning distributed particle swarm optimization for large-scale optimization. IEEE Trans Cybern 51(3):1175–1188
86. Bonyadi MR, Li X, Michalewicz Z (2014) A hybrid particle swarm with a time-adaptive topology for constrained optimization. Swarm Evol Comput 18:22–37
87. Brest J, Boskovic B, Zamuda A, Fister I, Maucec MS (2012) Self-adaptive differential evolution algorithm with a small and varying population size. In: IEEE congress on evolutionary computation, pp 1–8
88. Akbari MA, Zare M, Azizipanah-abarghooee R, Mirjalili S, Deriche M (2022) The cheetah optimizer: a nature-inspired metaheuristic algorithm for large-scale optimization problems. Sci Rep 12(1):10953
89. Chakraborty S, Saha AK, Chakraborty R, Saha M (2021) An enhanced whale optimization algorithm for large scale optimization problems. Knowl Based Syst 233:107543