Cross-Generation Elites Guided Particle Swarm Optimization for Large Scale Optimization

[1]Han-Yu Xie, [1,2]Qiang Yang, [3]Xiao-Min Hu and [2*]Wei-Neng Chen
[1]School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
[2]School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
[3]School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China

***Abstract*—Elites have been widely used in many evolutionary algorithms. However, only elites in current generation are utilized to guide the learning/updating of particles/individuals in existing algorithms. Usually, elites in different generations are different and elites in the past generations may contain experienced knowledge and thus may be helpful for guiding particles/individuals to promising areas. Inspired from this, we propose a Cross-generation Elites Guided Particle Swarm Optimizer in this paper. Specifically, the swarm in current generation is divided into two separate sets: the elite set containing the top best particles and the non-elite set consisting of the rest particles. Since these elite particles are the most promising ones in the current generation, we remain these elites unchanged and let them directly enter next generation. Then the rest non-elite particles are updated through learning from elites in both the current generation and the last generation. Through this, a potential balance between exploration and exploitation can be achieved. Particularly, the proposed algorithm is applied to deal with large scale optimization, which is very challenging and difficult and has received a lot of attention in recent years. Extensive experiments are conducted on two sets of large scale benchmark functions and experimental results verify the competitive effectiveness and efficiency of the proposed algorithm in comparison with several state-of-the-art large scale evolutionary algorithms.**

Keywords—Cross-Generation Elites; Elites; Partilce Swarm Optimization; Large Scale Optimization; Numerical Optimization.

I. INTRODUCTION

Since Particle Swarm Optimization (PSO) [1],[2] was first proposed by Kennedy and Eberhart in 1995, it has received plenty of attention in recent years and thus many PSO variants [3-8] have been proposed and applied to solve real-world problems [9],[10].

Generally, particles in the classical PSO have two properties: the particles' positions $\boldsymbol{X}$ and their velocities $\boldsymbol{V}$, which are updated as follows:

$$v_i^d \leftarrow wv_i^d + c_1r_1(pbest_i^d - x_i^d) + c_2r_2(gbest^d - x_i^d) \quad (1)$$

$$x_i^d \leftarrow x_i^d + v_i^d \quad (2)$$

where the ith particle is represented by $\boldsymbol{X}_i$=$[x_i^1,\ldots,x_i^d,\ldots,x_i^D]$ and $\boldsymbol{V}_i$=$[v_i^1,\ldots,v_i^d,\ldots,v_i^D]$with D denoting the dimension size. $\boldsymbol{pbest}_i$=$[pbest_i^1,\ldots,pbest_i^d,\ldots,pbest_i^D]$ is the personal best position of the ith particle, and the global best position of the swarm is $\boldsymbol{gbest}$=$[gbest^1,\ldots,gbest^d,\ldots,gbest^D]$. As for the parameters, w is termed as the inertia weight [11], c_1 and c_2 are two acceleration coefficients [1], and r_1 and r_2 are two uniformly random numbers within [0,1]. Kennedy and Eberhart [2] have taken the second part and the third part in the right of Eq. (1) as the cognitive component and the social component, respectively.

The classical PSO has been demonstrated to successfully solve low dimensional problems, especially unimodal problems. However, when it comes to complicated multimodal problems or high dimensional problems, its efficiency and effectiveness degrades drastically due to the serious loss of diversity, which results from the greedy attraction of the global best position ***gbest***. Thus, to locate the global optima of different problems efficiently, researchers have devoted a lot of attention to putting forward more effective learning or updating strategies to aid PSO to escape from local optima.

At first, since the global best position ***gbest*** is too greedy, researchers proposed to utilize the neighbor best position ***lbest*** to replace ***gbest*** in Eq. (1) to update particles through utilizing different topologies [12],[13]. Then, Chen *et.al.* transplanted the aging mechanism into PSO, leading to a PSO with aging leader and challenger (ALC-PSO) [6]; Based on Gaussian or Cauchy distributions, bare bone PSO (BBPSO) was developed by sampling the positions of particles [4],[14],[15]; Through recording and estimating the distribution of each particle's historical ***pbest***s, a composite PSO named HM-PSO was brought up [16].

Subsequently, to further enhance diversity, some PSO variants came up with generated exemplars, which are constructed by some kinds of learning strategies. The two most representatives are comprehensive learning PSO (CLPSO) [5] and orthogonal learning PSO (OLPSO) [17]. The former constructs exemplars through a comprehensive learning strategy, which selects each dimension of the exemplar from all particles' ***pbest***, while the latter adopts orthogonal experimental design on ***pbest*** and ***gbest*** (or ***lbest***) to obtain more efficient exemplars for each particle. Ren *et.al.* proposed a scatter learning PSO algorithm (SLPSOA) [18] by designing an exemplar pool composed of a fixed number of high-quality solutions. A genetic learning PSO (GL-PSO) [19] was put forward through using other EAs to generate

*** Corresponding Author (**Email: cwnraul634@aliyun.com **)**
This work was supported under the National Natural Science Foundation of China (NSFC) Grant Nos. 61622206, 61379061, 61332002, the Natural Science Foundation of Guangdong No. 2015A030306024, the Pearl River New Star of Science and Technology No. 201506010002 and the Guangdong Special Support Program No. 2014TQ01X550.

exemplars for particles.

Recently, elites, which are defined as the top best individuals in the population, have been intensively utilized to aid EAs including PSO to solve complicated problems [20-28], which will be detailed in next section. Observing these elite-guided EAs, we find that only elites in current generation are utilized. Since elites in different generations may be different and elites in past generations may contain experienced knowledge and thus may be helpful for particles to approach to promising areas, we propose a Cross-generation Elites Guided PSO (CEGPSO), which utilizes elites in both the current generation and the last generation to guide the learning of particles. Through this, a potential balance between exploration and exploitation can be achieved.

In particular, the proposed CEGPSO is applied to solve large scale optimization problems, which is very challenging and difficult to optimize, due to the exponentially increased search space and the rapidly increased number of local optima. These challenges lead to great degradation of traditional EAs, which though are very efficient in low dimensional space. The related work on large scale optimization will also be elucidated in next section.

Extensive experiments are conducted on CEC'2010 and CEC'2013 large scale benchmark functions to verify the effectiveness of the proposed CEGPSO in comparison with several state-of-the-art large scale algorithms.

The rest of this paper is organized as follows. In Section II, related work on both elite-guided EAs and large scale optimization are reviewed. Section III elaborates the proposed CEGPSO in details, following which is the experiments in Section IV to verify the effectiveness and efficiency of CEGPSO in dealing with large scale optimization. Finally, Section V concludes this paper.

II. Related Work

Without loss of generality, in this paper, we consider minimization problems formulated as follows:

$$\min f(\boldsymbol{x}) \qquad \boldsymbol{x} \in \boldsymbol{R}^D \tag{3}$$

where $f(\boldsymbol{x})$ is the function to be minimized and $\boldsymbol{x}$ is the variable vector containing D dimensions. In this paper, the function value is considered as the fitness value of a particle.

A. Elites Guided EAs

Elites are defined as the top best individuals in the population. They usually contain the most promising information to guide the learning or updating of particles or individuals. Currently, the elite mechanism is usually utilized in multi-objective optimization algorithms (MOEAs) [20],[23-25],[29],[30] to evolve the population, especially dominance-based MOEAs, such as NSGA-II [31].

The two most representatives that utilize elites to evolve the population in single objective optimization are estimation of distribution algorithms (EDA) [32],[33] and a DE variant named JADE [34] respectively. In most of the current EDAs, the distribution of the population is estimated based on the top best individuals in the current generation, namely the elites. JADE is a popular DE variant, which uses a mutation strategy called DE/current-to-pbest/1 to update individuals. In this strategy, the top p best individuals responsible for the generation of the offspring are just the elites in the current population.

Subsequently, Cui *et. al.* [26] developed an artificial bee colony algorithm with a depth-first search strategy, which not only exploits the information of the elite solutions but also employs the current best solution in the onlooker bee phase. In [21], a Genetic Algorithm (GA) variant was proposed by employing a hybrid memory and random immigrants scheme and a hybrid elitism and random immigrants scheme to handle dynamic optimization problems. In [22], an elite group guided quantum-inspired evolutionary algorithm (EQIEA) was brought up through introducing an elite group guidance updating strategy to solve knapsack problems. An elite-guided binary DE (EGBDE) in [27] and an elite-group guided DE (EMGDE) in [28] were respectively developed through using elites to guide the update of individuals to solve discrete and continuous optimization problems.

Observing these elite-guided EAs, we find that only elites in current generation are utilized. In general, elites in different generations are different and elites in the past generations may contain experienced knowledge and thus may be helpful for guiding particles to promising areas. Inspired from this, we develop a Cross-generation Elites Guided PSO (CEGPSO) through using elites in both the current generation and the last generation to guide the learning of particles.

B. Large Scale Optimization

When the dimension size of problems defined in Eq. (3) increases, the problems become more and more difficult to optimize due to the explosively increased search space and the exponentially increased number of local optima. Under such harsh environment, traditional EAs [4],[18],[35] usually drastically lose their efficiency and effectiveness.

Thus, to deal with high dimensional problems efficiently, researchers attempted to seek solutions in two aspects: 1) cooperative coevolution (CC) approaches, which divide the whole dimensions into several groups and evolve each dimension group separately; and 2) new learning or updating strategies, which evolve all dimensions as a whole like traditional EAs. Since the proposed method is a PSO variant, we mainly review PSO variants in dealing with large scale optimization in this section.

Utilizing the divide-and-conquer technique, Potter [36] proposed a cooperative coevolution (CC) framework to decompose problems into smaller sub-problems and then evolve them separately. Such a framework provides a promising approach for large scale optimization. Hereafter, researchers have embedded different EAs into this framework, leading to different CCEAs, such as cooperative co-evolutionary PSO (CCPSO) [37],[38], and cooperative co-evolutionary DE (DECC) [39-41].

In [37], CCPSO-S_K was developed by randomly dividing the whole D dimensions into D/K subcomponents and then each subcomponent is separately optimized by the classical PSO. Further, CCPSO- H_K was developed through alternatively using the classical PSO and CCPSO-S_K to update the swarm. To deal with the dilemma that the optimal number of subcomponents may be different for different problems, Li and Yao [38] proposed CCPSO2 by designing a group size pool containing different group sizes. When ***gbest*** is not updated, a group size is randomly selected from the pool, resulting in a dynamic decomposition strategy.

Since CCEAs optimize each subcomponent separately, the ideal decomposition strategy should put interdependent variables into the same group, while place independent variables into different groups [38],[39]. Currently, the most accurate grouping strategies for static variable dependency are differential grouping (DG) [39] and its variants, XDG [40] and GDG [41]. Though CCEAs show great potential in large scale optimization, they usually needs a lot of function evaluations to obtain satisfactory performance, due to optimizing each sub-component separately, especially when the number of groups is large. Besides, to get a good decomposition on dimensions, the decomposition technique usually needs a large number of function evaluations, which results in great reduction in the number of function evaluations used for evolution [39-41].

To alleviate the above situation, researchers attempt to develop new learning strategies for traditional EAs, which treat the problems as a whole like the classical PSO. Usually, special techniques are needed in the learning strategies to preserve high diversity, which is very important for high dimensional optimization [42].

In [43],[44], a dynamic multi-swarm PSO (DMS-PSO) was proposed by randomly dividing the whole swarm into multiple small sub-swarms. Then, a local version PSO is utilized to evolve each sub-swarms. Recently, in [45], [42], [46], a novel competitive learning strategy was put forward. By combining this learning strategy with different exemplar selection methods, different optimizers were developed. First, a multi-swarm PSO based on feedback evolution (FBE) [45] was proposed, where pairwise competition is performed between two particles randomly selected from two swarms, and then the loser is updated by a convergence strategy, while the winner is updated through a mutation strategy. Subsequently, a competitive swarm optimizer (CSO) [42] was developed, where the pairwise competition is executed among particles in a single swarm, and only the loser is updated, while the winner enters the next generation directly. The update of the loser is presented as follows:

$$v_l^d \leftarrow r_1 v_l^d + r_2(x_w^d - x_l^d) + \phi r_3(\bar{x}^d - x_l^d) \tag{4}$$

$$x_l^d \leftarrow x_l^d + v_l^d \tag{5}$$

where $\boldsymbol{X}_l$=$[x_l^1,\ldots,x_l^d,\ldots x_l^D]$ and $\boldsymbol{V}_l$=$[v_l^1,\ldots,v_l^d,\ldots v_l^D]$ are the position and speed of the loser respectively; the position of the winner is $\boldsymbol{X}_w$=$[x_w^1,\ldots,x_w^d,\ldots x_w^D]$ and $\bar{\boldsymbol{x}}$=$[\bar{x}^1,\ldots,\bar{x}^d,\ldots,\bar{x}^D]$ is the mean position of the swarm, r_1, r_2, and r_3 are three random variables ranging within [0,1] and ϕ is the control parameter in charge of the influence of $\bar{\boldsymbol{x}}$.

Further, a social learning PSO (SL-PSO) [46] was developed, where all particles are sorted according to their fitness values. Then, for each particle, the first exemplar is randomly selected from all better particles, while the second exemplar is also the mean position of the whole swarm.

In this paper, to solve large scale optimization efficiently, we propose CEGPSO through utilizing elites in both the current generation and the last generation to guide the learning of particles, which will be elaborated in next section.

III. The Proposed Algorithm

Elites are usually different in different generations and elites in the past generations may contain experienced knowledge and thus may be helpful in guiding particles to promising areas. Inspired from this, we develop a cross-generation elites guided PSO (CEGPSO) to deal with large scale optimization. Specifically, elites in both the current generation and the last generation are utilized to guide the learning of particles.

In particular, in this paper, elites are defined as the top *M* best particles in the swarm. Before evolution, the swarm is partitioned into two separate sets: the elite set ***ES*** containing the top *M* best particles and the non-elite set ***NES*** containing the rest particles. Since elites are usually the most promising particles in the swarm, we remain these elites unchanged and let them directly enter next generation. Thus, in each generation, only particles in ***NES*** are updated.

In this paper, we use elites in two consecutive generations to guide the learning of particles. Thus, two sets of elites exist, namely $\boldsymbol{ES}_G$ and $\boldsymbol{ES}_{G-1}$, with the former denoting the elites in the current generation, while the latter representing the elites in the last generation. To update the *i*th particle in $\boldsymbol{NES}_G$, two elites are first randomly selected with one randomly selected from $\boldsymbol{ES}_G$ ($\boldsymbol{X}_{el_{rand},\,G}$) and the other selected from $\boldsymbol{ES}_{G\text{-}1}$ ($\boldsymbol{X}_{el_{rand},\,G\text{-}1}$). Then, these two elites compete with each other and the winner acts as the first exemplar to replace ***pbest*** in Eq. (1) and the loser acts as the second exemplar to replace ***gbest*** in Eq. (1) to guide the learning of particles. Specifically, the update formula of particles in $\boldsymbol{NES}_G$ is presented as follows:

$$v_i^d \leftarrow r_1 v_i^d + r_2(x_{el_1}^d - x_i^d) + \phi r_3(x_{el_2}^d - x_i^d) \tag{6}$$

$$x_i^d \leftarrow x_i^d + v_i^d \tag{7}$$

$$\begin{cases} \begin{cases} \boldsymbol{X}_{el_1} = \boldsymbol{X}_{el_{rand},G} \\ \boldsymbol{X}_{el_2} = \boldsymbol{X}_{el_{rand},G-1} \end{cases}, & \text{if } f(\boldsymbol{X}_{el_{rand},G}) < f(\boldsymbol{X}_{el_{rand},G-1}); \\ \begin{cases} \boldsymbol{X}_{el_1} = \boldsymbol{X}_{el_{rand},G-1} \\ \boldsymbol{X}_{el_2} = \boldsymbol{X}_{el_{rand},G} \end{cases}, & \text{otherwise} \end{cases} \tag{8}$$

where $\boldsymbol{X}_i$=$[x_i^1,\ldots,x_i^d,\ldots x_i^D]$ and $\boldsymbol{V}_i$=$[v_i^1,\ldots,v_i^d,\ldots v_i^D]$ respectively denote the position and speed of the *i*th particle in $\boldsymbol{NES}_G$. $\boldsymbol{X}_{el_1}$=$[x_{el_1}^1,\ldots,x_{el_1}^d,\ldots,x_{el_1}^D]$ and $\boldsymbol{X}_{el_2}$=$[x_{el_2}^1,\ldots,x_{el_2}^d,\ldots,x_{el_2}^D]$ are the first and second elites respectively used to guide the learning of particles. These two elites are determined by Eq. (8) where $\boldsymbol{X}_{el_{rand},G}$ is an elite randomly selected from $\boldsymbol{ES}_G$ while $\boldsymbol{X}_{el_{rand},G\text{-}1}$ is an elite randomly selected from $\boldsymbol{ES}_{G\text{-}1}$. As for the parameters, r_1, r_2, and r_3 are three random variables ranging within [0,1] and ϕ is a control parameter in charge of the influence of the second elite.

In Eq. (6), the second part in the right is mainly responsible for guiding particles to promising areas, while the third part is mainly to enlarge the diversity of the swarm, so that premature convergence can be avoided. Thus, the better one between the two selected elites acts as the first exemplar in

Eq. (6) while the worse one acts as the second exemplar, because the better elite usually has more powerful exploitation ability while the worse one generally has more powerful exploration ability.

Even though Eq. (6) ~ Eq. (8) have already characterized the proposed CEGPSO, some detailed techniques should be mentioned here:

1) In the initial stage, $\boldsymbol{ES}_0$ is set to be empty. Thus, for the update of particles in $\boldsymbol{NES}_1$ in the first generation, only one set of elites exists, namely $\boldsymbol{ES}_1$. Therefore, for particles in $\boldsymbol{NES}_1$, the two random elites are both selected from $\boldsymbol{ES}_1$ with the winner acting as the first exemplar, while the loser acting as the second exemplar.
2) As for the selection of elites, in this paper, we utilize the roulette wheel selection strategy to respectively select one elite from $\boldsymbol{ES}_G$ and $\boldsymbol{ES}_{G-1}$. The probability of each elite in $\boldsymbol{ES}_G$ is computed as follows:
$$p_{el_j} = \frac{M - rk_j + 1}{\sum_{k=1}^{M} k} \tag{9}$$
where p_{el_j} is the probability of the jth elite, M is the elite set ($\boldsymbol{ES}_G$) size, and rk_j is the ranking of the jth elite, which is determined by its fitness value.
3) In this paper, we adopt the following repair mechanism when the position of one particle is out of the bounds:
$$x_i^d = \begin{cases} x_{\max}^d - (x_i^d - x_{\max}^d), & if \quad x_i^d > x_{\max}^d \\ x_{\min}^d + (x_{\min}^d - x_i^d), & if \quad x_i^d < x_{\min}^d \end{cases} \tag{10}$$
where x_{max}^d and x_{min}^d are the upper and lower bounds of the dth dimension respectively.

In CEGPSO, the elites are preserved and remain unchanged. During evolution, the elites in the current generation are generally much better than those in the last generation as a whole. Therefore, we can see that during evolution, the elites in the swarm become better and better, and finally these elites could converge together at the global optima or local optima of the problems to be optimized.

Overall, the complete framework of the proposed CEGPSO is presented in **Algorithm 1**. Comprehensively, we can see that CEGPSO is simple to implement due to its maintenance of the classical framework of PSO. In terms of the time complexity, compared with the classical PSO, CEGPSO only needs extra $O(NP\log(NP))$ to sort the population according to fitness values and then gets the elites in the current generation. As for the space complexity, CEGPSO needs much less space than the classical PSO, because it only needs $O(M*D)$ space to keep the elites in the last generation, which is much smaller than $O(NP*D)$ to store the personal best position of each particle in the classical PSO. Together, we can see that CEGPSO is both time and space efficient.

Algorithm 1: The pseudo code of CEGPSO

Input: Swarm size NP, Maximum number of function evaluations Max_Fes, Number of elites M, Control parameter ϕ;

```
Fes = 0, G=0 and set ES0 to be an empty set;
Initialize the swarm randomly;
Evaluate particles in the swarm P;
Fes = Fes + NP;
While Fes < Max_Fes
    G=G+1;
    Sort particles in the swarm and put the indexes of the top best M
      particles into ESG and the others into NESG;
    For i = 1: | NESG |
        If G==1
            Randomly select two elites from ESG: Xel1 and Xel2;
        Else
            Randomly select one elite from ESG: Xel1 and one elite from
              ESG-1: Xel2;
        End If
        If f(Xel2)<f(Xel1)
            Swap Xel1 and Xel2;
        End If
        Update this particle according to Eq. (6) and Eq. (7);
        Repair the position of this particle according to Eq. (10);
        Evaluate this particle;
        Fes= Fes +1;
    End For
    Put the elite particles in ESG into ESG-1;
End While
```

Output: The global best fitness and the global best individual.

IV. Experimental Studies

A. Experiment Setup

1) Benchmark Functions

To verify the performance of the proposed CEGPSO, we conduct experiments on CEC'2010 [47] and CEC'2013 [48] benchmark functions with 1000 dimensions. Functions in the latter set are the extensions of those in the former set, but they are much more difficult to optimize than the former ones. For the detailed description of these benchmark functions, readers can be referred to [47],[48].

2) Compared Algorithms

To comprehensively demonstrate the performance of CEGPSO, several state-of-the-art evolutionary algorithms dealing with large scale optimization are selected to make fair comparisons. These compared algorithms are 1) CSO [42], SL-PSO [46] and DMS-L-PSO [43], which evolve all variables together like traditional EAs; 2) CCPSO2 [38], DECC-G [49], MLCC [50], and DECC-DG [39], which are all CCEAs and thus decompose the dimensions into several groups and evolve each group separately. These two kinds of EAs in coping with large scale optimization are selected, so that a comprehensive comparison can be achieved

3) Parameter Settings

As for the parameter settings, the main parameter settings of CEGPSO are listed in Table I. For the compared algorithms, the main parameters are set according to the corresponding papers.

TABLE I
PARAMETER SETTINGS FOR CEGPSO

NP	M	Φ
500	0.2NP	0.6

In terms of the maximum number of function evaluations, without otherwise stated, it is set as 3×10^6, at which the comparison results are reported in Table II and Table III.

Table II

COMPARISON RESULTS OF THE COMPARED ALGORITHMS ON 1000-*D* CEC'2010 FUNCTIONS WITH 3×10^{6} FITNESS EVALUTIONS

F	Quality	CEGPSO	CSO	SL-PSO	DMS-L-PSO	CCPSO2	DECC-G	MLCC	DECC-DG
F_1	Median	2.10E-21	4.40E-12	7.90E-18	1.61E+07	7.80E-01	3.53E-07	1.66E-14	1.42E+02
	Mean	2.06E-21	4.50E-12	8.73E-18	1.63E+07	2.96E+00	3.54E-07	8.65E-13	1.88E+04
	Std	2.86E-22	5.94E-13	3.30E-18	1.41E+06	6.68E+00	1.44E-07	2.97E-12	4.66E+04
	p-value	-	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$
F_2	Median	1.47E+03	7.33E+03	1.93E+03	5.53E+03	4.25E+00	1.32E+03	2.43E+00	4.46E+03
	Mean	1.48E+03	7.42E+03	1.93E+03	5.45E+03	4.30E+00	1.33E+03	2.89E+00	4.43E+03
	Std	7.67E+01	2.86E+02	1.12E+02	5.38E+02	1.11E+00	2.55E+01	1.52E+00	1.87E+02
	p-value	-	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	3.02E-11^{-}	1.21E-10^{-}	3.02E-11^{-}	**3.02E-11**$^{+}$
F_3	Median	4.13E-14	2.53E-09	1.85E+00	1.56E+01	4.16E-03	1.14E+00	6.24E-10	1.66E+01
	Mean	4.13E-14	2.60E-09	1.88E+00	1.56E+01	4.51E-03	1.10E+00	2.10E-07	1.66E+01
	Std	1.81E-15	2.62E-10	3.30E-01	1.08E-01	1.66E-03	3.35E-01	1.12E-06	3.02E-01
	p-value	-	**1.46E-11**$^{+}$	**1.46E-11**$^{+}$	**1.46E-11**$^{+}$	**1.46E-11**$^{+}$	**1.46E-11**$^{+}$	**1.46E-11**$^{+}$	**1.46E-11**$^{+}$
F_4	Median	2.88E+11	7.26E+11	3.04E+11	4.32E+11	1.45E+12	2.46E+13	1.78E+13	5.08E+12
	Mean	2.93E+11	7.25E+11	2.99E+11	4.42E+11	1.70E+12	2.59E+13	1.71E+13	5.22E+12
	Std	7.12E+10	1.23E+11	7.16E+10	8.05E+10	1.04E+12	8.14E+12	5.47E+12	1.89E+12
	p-value	-	**3.34E-11**$^{+}$	6.10E-01$^{=}$	**1.31E-08**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$
F_5	Median	2.86E+08	2.00E+06	3.29E+07	9.35E+07	3.76E+08	2.50E+08	5.11E+08	1.52E+08
	Mean	2.83E+08	2.86E+06	3.17E+07	9.25E+07	4.14E+08	2.69E+08	4.99E+08	1.55E+08
	Std	1.28E+07	1.79E+06	6.21E+06	9.04E+06	1.38E+08	6.84E+07	1.07E+08	2.15E+07
	p-value	-	3.02E-11^{-}	3.02E-11^{-}	3.02E-11^{-}	**6.28E-06**$^{+}$	2.42E-02^{-}	**5.57E-10**$^{+}$	3.02E-11^{-}
F_6	Median	4.00E-09	8.23E-07	2.15E+01	3.66E+01	1.97E+07	4.71E+06	1.97E+07	1.64E+01
	Mean	4.12E-09	8.21E-07	2.08E+01	3.78E+01	1.71E+07	5.00E+06	1.78E+07	1.63E+01
	Std	6.55E-10	2.68E-08	2.63E+00	1.21E+01	5.20E+06	1.03E+06	4.37E+06	3.45E-01
	p-value	-	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$
F_7	Median	1.19E+00	2.04E+04	4.06E+04	3.47E+06	2.67E+06	6.57E+08	1.15E+08	9.20E+03
	Mean	1.33E+00	2.01E+04	6.49E+04	3.47E+06	2.06E+08	8.14E+08	1.51E+08	1.41E+04
	Std	6.98E-01	3.86E+03	5.60E+04	1.16E+05	4.31E+08	5.40E+08	1.45E+08	1.26E+04
	p-value	-	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$
F_8	Median	2.99E+07	3.87E+07	7.43E+06	2.02E+07	2.00E+07	9.06E+07	8.82E+07	1.62E+07
	Mean	2.99E+07	3.87E+07	7.81E+06	2.03E+07	4.13E+07	8.56E+07	6.59E+07	2.75E+07
	Std	2.39E+05	6.81E+04	1.56E+06	1.88E+06	3.84E+07	2.64E+07	3.40E+07	2.63E+07
	p-value	-	**3.02E-11**$^{+}$	3.02E-11^{-}	3.02E-11^{-}	1.86E-01$^{=}$	**8.48E-09**$^{+}$	**6.77E-05**$^{+}$	3.99E-04^{-}
F_9	Median	4.24E+07	7.05E+07	3.22E+07	2.08E+07	1.14E+08	4.35E+08	2.48E+08	5.52E+07
	Mean	4.20E+07	7.03E+07	3.30E+07	2.08E+07	1.02E+08	4.40E+08	2.43E+08	5.59E+07
	Std	3.36E+06	5.73E+06	4.46E+06	1.58E+06	3.30E+07	4.87E+07	2.16E+07	6.45E+06
	p-value	-	**3.02E-11**$^{+}$	7.77E-09^{-}	3.02E-11^{-}	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**8.15E-11**$^{+}$
F_{10}	Median	1.55E+03	9.59E+03	2.60E+03	5.09E+03	5.14E+03	1.02E+04	3.97E+03	4.47E+03
	Mean	1.56E+03	9.60E+03	2.56E+03	5.24E+03	5.09E+03	1.03E+04	4.24E+03	4.49E+03
	Std	6.22E+01	7.67E+01	2.17E+02	4.26E+02	7.81E+02	3.13E+02	1.45E+03	1.29E+02
	p-value	-	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$
F_{11}	Median	1.95E+01	3.80E-08	2.30E+01	1.68E+02	1.98E+02	2.59E+01	1.98E+02	1.02E+01
	Mean	1.56E+01	4.02E-08	2.32E+01	1.68E+02	1.98E+02	2.59E+01	1.98E+02	1.02E+01
	Std	7.84E+00	5.12E-09	2.10E+00	1.90E+00	2.12E+00	1.73E+00	1.12E+00	8.71E-01
	p-value	-	9.51E-06^{-}	**7.38E-10**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	6.77E-05^{-}
F_{12}	Median	2.20E+04	4.23E+05	1.31E+04	2.83E+01	2.78E+04	9.69E+04	1.01E+05	2.58E+03
	Mean	2.21E+04	4.37E+05	1.75E+04	2.39E+01	3.39E+04	9.55E+04	1.03E+05	2.84E+03
	Std	1.86E+03	6.22E+04	9.07E+03	9.88E+00	1.19E+04	9.55E+03	1.57E+04	1.08E+03
	p-value	-	**3.02E-11**$^{+}$	1.37E-03^{-}	3.02E-11^{-}	**6.53E-07**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	3.02E-11^{-}
F_{13}	Median	5.89E+02	5.47E+02	8.48E+02	1.03E+05	1.36E+03	4.59E+03	2.12E+03	5.06E+03
	Mean	6.13E+02	6.29E+02	9.59E+02	1.05E+05	1.34E+03	5.96E+03	4.22E+03	6.27E+03
	Std	1.72E+02	2.32E+02	3.74E+02	6.18E+04	1.72E+02	4.16E+03	4.70E+03	3.65E+03
	p-value	-	8.19E-01$^{=}$	**2.15E-06**$^{+}$	**3.02E-11**$^{+}$	**9.92E-11**$^{+}$	**3.02E-11**$^{+}$	**4.08E-11**$^{+}$	**3.02E-11**$^{+}$
F_{14}	Median	1.24E+08	2.52E+08	8.45E+07	1.25E+07	3.42E+08	9.72E+08	5.71E+08	3.46E+08
	Mean	1.24E+08	2.49E+08	8.41E+07	1.19E+07	3.06E+08	9.78E+08	5.70E+08	3.42E+08
	Std	8.13E+06	1.53E+07	6.31E+06	1.62E+06	1.19E+08	7.52E+07	5.50E+07	2.42E+07
	p-value	-	**3.02E-11**$^{+}$	3.02E-11^{-}	3.02E-11^{-}	**4.98E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$
F_{15}	Median	1.08E+04	1.01E+04	1.12E+04	5.48E+03	1.04E+04	1.24E+04	8.67E+03	5.86E+03
	Mean	1.08E+04	1.01E+04	1.12E+04	5.54E+03	1.08E+04	1.23E+04	8.90E+03	5.86E+03
	Std	9.75E+01	5.23E+01	8.65E+01	3.46E+02	1.35E+03	8.24E+02	2.07E+03	1.05E+02
	p-value	-	3.02E-11^{-}	**3.02E-11**$^{+}$	3.02E-11^{-}	4.68E-02^{-}	**3.02E-11**$^{+}$	1.09E-05^{-}	3.02E-11^{-}
F_{16}	Median	2.15E+00	5.75E-08	2.21E+01	3.18E+02	3.97E+02	6.92E+01	3.96E+02	7.50E-13
	Mean	2.40E+00	5.89E-08	2.51E+01	3.17E+02	3.96E+02	6.96E+01	3.81E+02	7.53E-13
	Std	1.89E+00	5.61E-09	1.16E+01	2.04E+00	5.73E-01	6.43E+00	5.76E+01	6.25E-14
	p-value	-	1.10E-06^{-}	**3.00E-11**$^{+}$	**3.00E-11**$^{+}$	**3.00E-11**$^{+}$	**3.00E-11**$^{+}$	**3.00E-11**$^{+}$	1.10E-06^{-}
F_{17}	Median	2.06E+05	2.22E+06	8.92E+04	4.75E+01	8.99E+04	3.11E+05	3.47E+05	4.02E+04
	Mean	2.06E+05	2.20E+06	9.00E+04	4.29E+01	1.25E+05	3.11E+05	3.49E+05	4.03E+04
	Std	1.37E+04	1.55E+05	1.58E+04	1.15E+01	5.25E+04	2.24E+04	3.11E+04	2.29E+03
	p-value	-	**3.02E-11**$^{+}$	3.02E-11^{-}	3.02E-11^{-}	5.46E-09^{-}	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	3.02E-11^{-}
F_{18}	Median	1.81E+03	1.76E+03	2.66E+03	2.50E+04	3.10E+03	3.54E+04	1.59E+04	1.47E+10
	Mean	2.16E+03	1.73E+03	2.77E+03	2.43E+04	3.07E+03	3.83E+04	1.81E+04	1.47E+10
	Std	8.05E+02	5.22E+02	8.33E+02	1.10E+04	2.45E+02	1.53E+04	9.48E+03	2.03E+09
	p-value	-	4.21E-02^{-}	**1.77E-03**$^{+}$	**3.02E-11**$^{+}$	**1.86E-06**$^{+}$	**3.02E-11**$^{+}$	**4.50E-11**$^{+}$	**3.02E-11**$^{+}$
F_{19}	Median	5.72E+06	1.00E+07	5.11E+06	2.03E+06	1.51E+06	1.14E+06	2.04E+06	1.75E+06
	Mean	5.72E+06	1.01E+07	5.10E+06	2.02E+06	1.52E+06	1.14E+06	2.04E+06	1.75E+06
	Std	4.92E+05	5.64E+05	7.05E+05	1.41E+05	7.10E+04	6.23E+04	1.42E+05	1.10E+05
	p-value	-	**3.02E-11**$^{+}$	1.32E-04^{-}	3.02E-11^{-}	3.02E-11^{-}	3.02E-11^{-}	3.02E-11^{-}	3.02E-11^{-}
F_{20}	Median	1.54E+03	9.85E+02	1.79E+03	9.82E+02	2.10E+03	4.34E+03	2.27E+03	6.53E+10
	Mean	1.56E+03	1.05E+03	1.85E+03	9.87E+02	2.11E+03	4.58E+03	2.30E+03	6.41E+10
	Std	2.27E+02	1.49E+02	2.59E+02	1.40E+01	1.79E+02	8.25E+02	2.26E+02	6.97E+09
	p-value	-	5.57E-10^{-}	**1.39E-06**$^{+}$	3.02E-11^{-}	**1.55E-09**$^{+}$	**3.02E-11**$^{+}$	**3.16E-10**$^{+}$	**3.02E-11**$^{+}$
w/l/t		-	13/6/1	12/7/1	11/9/0	15/4/1	17/3/0	17/3/0	12/8/0

Table III

COMPARISON RESULTS OF THE COMPARED ALGORITHMS ON 1000-D CEC'2013 FUNCTIONS WITH 3×10^6 FITNESS EVALUTIONS

F	Quality	CEGPSO	CSO	SL-PSO	DMS-L-PSO	CCPSO2	DECC-G	MLCC	DECC-DG
F_1	Median	2.23E-21	7.78E-12	1.04E-17	1.97E+09	2.79E+01	2.06E-06	9.07E-14	6.03E+02
	Mean	2.38E-21	7.71E-12	1.09E-17	1.98E+09	4.11E+01	3.14E-06	8.60E-10	6.42E+03
	Std	4.67E-22	1.31E-12	2.50E-18	1.27E+08	3.14E+01	4.27E-06	4.38E-09	1.81E+04
	p-value	-	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$
F_2	Median	1.62E+03	8.55E+03	2.13E+03	8.61E+03	3.48E+01	1.30E+03	3.57E+00	1.28E+04
	Mean	1.61E+03	8.55E+03	2.13E+03	8.65E+03	3.50E+01	1.31E+03	3.82E+00	1.27E+04
	Std	8.89E+01	2.65E+02	1.36E+02	4.88E+02	4.85E+00	3.63E+01	1.73E+00	7.20E+02
	p-value	-	**3.02E-11**$^{+}$	**3.34E-11**$^{+}$	**3.02E-11**$^{+}$	3.02E-11^{-}	3.02E-11^{-}	3.02E-11^{-}	**3.02E-11**$^{+}$
F_3	Median	2.16E+01	2.16E+01	2.16E+01	2.08E+01	2.00E+01	2.02E+01	2.00E+01	2.14E+01
	Mean	2.16E+01	2.16E+01	2.16E+01	2.08E+01	2.00E+01	2.02E+01	2.00E+01	2.14E+01
	Std	5.99E-03	6.15E-03	1.45E-02	1.66E-01	1.25E-04	6.18E-03	2.76E-04	1.45E-02
	p-value	-	7.73E-01$^{=}$	**2.68E-06**$^{+}$	3.02E-11^{-}	2.19E-11^{-}	3.02E-11^{-}	3.02E-11^{-}	3.02E-11^{-}
F_4	Median	7.72E+09	1.28E+10	4.54E+09	2.97E+11	3.20E+10	2.00E+11	1.99E+11	7.33E+10
	Mean	7.91E+09	1.32E+10	4.35E+09	2.93E+11	3.49E+10	2.35E+11	2.34E+11	7.70E+10
	Std	2.48E+09	2.54E+09	9.48E+08	7.25E+10	2.17E+10	1.22E+11	1.26E+11	2.82E+10
	p-value	-	**7.77E-09**$^{+}$	4.57E-09^{-}	**3.02E-11**$^{+}$	**1.56E-08**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$
F_5	Median	7.57E+05	6.04E+05	8.23E+05	3.92E+06	1.30E+07	8.44E+06	1.17E+07	5.81E+06
	Mean	7.39E+05	5.91E+05	8.41E+05	3.95E+06	1.40E+07	8.26E+06	1.27E+07	5.78E+06
	Std	9.55E+04	1.07E+05	1.75E+05	5.82E+05	4.81E+06	1.14E+06	3.46E+06	3.83E+05
	p-value	-	2.88E-06^{-}	**1.08E-02**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$
F_6	Median	1.06E+06	1.06E+06	1.06E+06	9.98E+05	1.05E+06	1.06E+06	1.05E+06	1.06E+06
	Mean	1.06E+06	1.06E+06	1.06E+06	1.00E+06	1.05E+06	1.06E+06	1.05E+06	1.06E+06
	Std	1.11E+03	1.10E+03	1.48E+03	5.20E+03	5.24E+03	1.84E+03	4.13E+03	1.07E+03
	p-value	-	**4.21E-02**$^{+}$	5.01E-01$^{=}$	3.02E-11^{-}	3.02E-11^{-}	1.37E-03^{-}	4.08E-11^{-}	**3.56E-04**$^{+}$
F_7	Median	4.22E+06	5.26E+06	1.40E+06	1.22E+09	1.29E+08	1.04E+09	1.15E+09	4.25E+08
	Mean	4.37E+06	5.88E+06	1.63E+06	1.37E+09	4.15E+08	1.04E+09	1.43E+09	4.78E+08
	Std	2.15E+06	2.58E+06	7.05E+05	7.64E+08	9.38E+08	4.48E+08	1.07E+09	1.92E+08
	p-value	-	**1.17E-02**$^{+}$	8.48E-09^{-}	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$
F_8	Median	7.53E+13	2.85E+14	9.97E+13	1.68E+14	8.16E+14	7.90E+15	8.18E+15	2.89E+15
	Mean	7.96E+13	2.60E+14	1.03E+14	2.79E+14	1.18E+15	7.50E+15	9.59E+15	3.57E+15
	Std	2.28E+13	5.87E+13	3.62E+13	5.09E+14	9.99E+14	3.18E+15	6.18E+15	1.85E+15
	p-value	-	**3.02E-11**$^{+}$	**1.56E-02**$^{+}$	**5.49E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$
F_9	Median	1.75E+08	5.79E+07	7.94E+07	3.50E+08	3.63E+09	5.86E+08	8.85E+08	4.95E+08
	Mean	1.72E+08	6.06E+07	8.25E+07	3.60E+08	3.76E+09	5.96E+08	9.55E+08	4.90E+08
	Std	4.45E+07	1.60E+07	2.03E+07	4.61E+07	1.02E+09	9.76E+07	2.92E+08	3.18E+07
	p-value	-	4.08E-11^{-}	2.87E-10^{-}	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$
F_{10}	Median	9.40E+07	9.40E+07	9.36E+07	9.11E+07	9.29E+07	9.30E+07	9.27E+07	9.45E+07
	Mean	9.40E+07	9.40E+07	9.25E+07	9.17E+07	9.30E+07	9.29E+07	9.27E+07	9.45E+07
	Std	3.05E+05	1.51E+05	1.67E+06	1.06E+06	7.01E+05	6.16E+05	6.07E+05	2.46E+05
	p-value	-	2.17E-01$^{=}$	5.56E-04^{-}	3.69E-11^{-}	8.84E-07^{-}	1.29E-09^{-}	1.61E-10^{-}	**3.65E-08**$^{+}$
F_{11}	Median	9.20E+11	9.35E+11	9.35E+11	9.44E+10	9.38E+11	1.26E+11	1.90E+11	3.81E+10
	Mean	9.22E+11	9.30E+11	9.33E+11	1.05E+11	9.37E+11	1.28E+11	2.28E+11	4.83E+10
	Std	6.57E+09	1.03E+10	1.46E+10	7.53E+10	1.53E+10	7.15E+10	1.53E+11	4.33E+10
	p-value	-	**4.64E-03**$^{+}$	**1.89E-04**$^{+}$	3.02E-11^{-}	**5.19E-07**$^{+}$	3.02E-11^{-}	3.02E-11^{-}	3.02E-11^{-}
F_{12}	Median	1.54E+03	1.04E+03	1.75E+03	5.22E+04	2.10E+03	4.19E+03	2.36E+03	1.68E+11
	Mean	1.55E+03	1.07E+03	1.78E+03	6.99E+04	2.10E+03	4.35E+03	2.49E+03	1.71E+11
	Std	1.66E+02	7.78E+01	1.74E+02	5.52E+04	1.78E+02	7.83E+02	7.51E+02	2.24E+10
	p-value	-	7.39E-11^{-}	**1.25E-05**$^{+}$	**3.02E-11**$^{+}$	**1.21E-10**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$
F_{13}	Median	8.74E+08	6.28E+08	4.59E+08	1.32E+10	3.21E+09	8.67E+09	9.94E+09	2.08E+10
	Mean	9.73E+08	6.67E+08	4.65E+08	1.34E+10	4.02E+09	9.35E+09	1.06E+10	2.05E+10
	Std	4.49E+08	2.45E+08	2.35E+08	6.58E+09	2.31E+09	2.78E+09	3.73E+09	5.53E+09
	p-value	-	1.06E-03^{-}	4.44E-07^{-}	**3.02E-11**$^{+}$	**1.46E-10**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$
F_{14}	Median	8.01E+08	3.15E+09	1.50E+08	2.21E+11	5.98E+10	1.28E+11	2.06E+11	1.56E+10
	Mean	1.34E+09	3.62E+09	3.28E+08	2.46E+11	9.10E+10	1.42E+11	2.21E+11	1.92E+10
	Std	1.31E+09	1.44E+09	5.17E+08	1.26E+11	8.53E+10	5.86E+10	8.54E+10	1.44E+10
	p-value	-	**9.06E-08**$^{+}$	7.09E-08^{-}	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	**9.92E-11**$^{+}$
F_{15}	Median	2.09E+07	7.72E+07	5.88E+07	1.54E+07	2.72E+06	1.13E+07	1.57E+07	9.52E+06
	Mean	2.30E+07	7.87E+07	5.86E+07	1.57E+07	4.75E+06	1.16E+07	1.61E+07	9.90E+06
	Std	9.24E+06	6.50E+06	6.11E+06	3.45E+06	5.07E+06	1.26E+06	1.90E+06	2.30E+06
	p-value	-	**3.02E-11**$^{+}$	**3.02E-11**$^{+}$	4.03E-03^{-}	7.38E-10^{-}	1.70E-08^{-}	8.31E-03^{-}	1.07E-09^{-}
w/l/t		-	9/4/2	8/6/1	10/5/0	10/5/0	9/6/0	9/6/0	12/3/0

Additionally, it is worth mentioning that all experiments are conducted 30 independent runs for statistical analysis. During the comparison between two algorithms, Wilcoxon rank sum test at a significance level of α=0.05 is conducted, and the corresponding p-values are reported. Based on the statistical test results, in Table II and Table III, the best results are highlighted in bold and the symbols "+", "−" and "=" above p-values indicate that CEGPSO is significantly better than, worse than and equivalent to the compared methods respectively. Specially, the last row in Table II and Table III, termed as "*w/l/t*" counts the number of functions where CEGPSO performs better than, worse than and similarly to the compared algorithms respectively.

4) Experimental Results

Table II and Table III present the comparison results between CEGPSO and the compared algorithms on the CEC'2010 benchmark set and the CEC'2013 benchmark set respectively. From these two tables, we can draw the following conclusions:

1) On the CEC'2010 set, the proposed CEGPSO outperforms the compared algorithms on at least 11 functions. Specifically, CEGPSO performs significantly better than CSO, SL-PSO and DMS-L-PSO on 13, 12 and 11 functions respectively. Besides, it is also much better than CCPSO2, DECC-G, MLCC and DECC-DG on 15, 17, 17 and 12 functions respectively. In particular, we find that CEGPSO shows its great superiority to CCPSO2, DECC-G and MLCC on this function set.
2) On the CEC'2013 set, whose functions are much more difficult to optimize than those in the former set, the developed CEGPSO is still significantly better than the compared algorithms on at least 8 functions. More specifically, CEGPSO is much superior to CSO, SL-PSO and DMS-L-PSO on 9, 8 and 10 functions respectively and significantly dominates CCPSO2, DECC-G, MLCC and DECC-DG on 10, 9, 9, and 12 functions respectively.
3) Together, we can see that compared with CCEAs, which decompose the high dimensional problems into smaller sub-problems, CEGPSO shows its great advantages on both CEC'2010 and CEC'2013 benchmark sets. In terms of large scale EAs that evolve all variable together, CEGPSO can achieve competitive or even much better performance on both benchmark function sets.

Overall, we can see that the proposed CEGPSO shows its great potential in dealing with large scale optimization. The great superiority of CEGPSO mainly results from the proposed cross-generation elites guided learning strategy. On one hand, each non-elite particle is guided by two elites randomly selected from the two elite sets between two consecutive generations. Since elites are usually the most promising particles in the population, the exploitation of each non-elite particle can be enhanced, which is beneficial for fast convergence. On the other hand, in general, after each generation, the elites are usually updated and thus the elites between two consecutive generations are probably different. Therefore, the diversity of the swarm is very high and thus the exploration of non-elite particles can be promoted, which is helpful for escaping from local traps. Comprehensively, a potential balance between exploration and exploitation can be achieved, which leads to the promising performance of CEGPSO in dealing with large scale optimization.

V. Conclusion

This paper has proposed a cross-generation elites guided particle swarm optimizer (CEGPSO) for large scale optimization. Specifically, this optimizer separates the swarm into two non-overlapping sets: the elite set and the non-elite set. Particles in the elite set remain unchanged and directly enter next generation, so that the promising information of the swarm can be preserved. Instead, for particles in the non-elite set, the elites in both the current generation and the last generation are utilized to guide the learning of them. Through this strategy, the elites become better and better and thus may converge to the global optima or local optima of problems.

Extensive experiments are conducted on CEC'2010 and CEC'2013 large scale benchmark sets. The statistical results demonstrate that the developed CEGPSO is promising and can obtain competitive performance in dealing with large scale optimization in comparison with state-of-the-art large scale EAs.

In particular, we find that the number of elites, namely M, has significant influence on the developed CEGPSO. Therefore, how to self-adaptively adjust this parameter forms a part of future work.

Reference

[1] R. C. Eberhart, and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. Int. Symp. MHS*, pp. 39-43, 1995.

[2] J. Kennedy, and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw*, pp. 1942-1948 vol.4, 1995.

[3] M. Hu, T. Wu, and J. D. Weir, "An adaptive particle swarm optimization with multiple adaptive methods," *IEEE Trans. Evol. Comput.,* vol. 17, no. 5, pp. 705-720, 2013.

[4] M. Campos, R. A. Krohling, and I. Enriquez, "Bare Bones Particle Swarm Optimization With Scale Matrix Adaptation," *IEEE Trans. Cybern.,* vol. 44, no. 9, pp. 1567-1578, 2014.

[5] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.,* vol. 10, no. 3, pp. 281-295, 2006.

[6] W.-N. Chen, J. Zhang, Y. Lin, N. Chen, Z.-H. Zhan, H. S.-H. Chung, Y. Li, and Y.-h. Shi, "Particle Swarm Optimization With an Aging Leader and Challengers," *IEEE Trans. Evol. Comput.,* vol. 17, no. 2, pp. 241-258, 2013.

[7] S. K. Chou, M. K. Jiau, and S. C. Huang, "Stochastic Set-Based Particle Swarm Optimization Based on Local Exploration for Solving the Carpool Service Problem," *IEEE Trans. Cybern.,* vol. PP, no. 99, pp. 1-13, 2016.

[8] W. N. Chen, J. Zhang, H. S. H. Chung, W. L. Zhong, W. G. Wu, and Y. h. Shi, "A Novel Set-Based Particle Swarm Optimization Method for Discrete Optimization Problems," *IEEE Trans. Evol. Comput.,* vol. 14, no. 2, pp. 278-300, 2010.

[9] P. Faria, J. Soares, Z. Vale, H. Morais, and T. Sousa, "Modified Particle Swarm Optimization Applied to Integrated Demand Response and DG Resources Scheduling," *IEEE Trans. Smart Grid,* vol. 4, no. 1, pp. 606-616, 2013.

[10] Y.-J. Gong, M. Shen, J. Zhang, O. Kaynak, W.-N. Chen, and Z.-H. Zhan, "Optimizing RFID network planning by using a particle swarm optimization algorithm with redundant reader elimination," *IEEE Trans. Ind. Informat,* vol. 8, no. 4, pp. 900-912, 2012.

[11] Y. Shi, and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Congr. Evol. Comput.*, pp. 69-73, 1998.

[12] J. Kennedy, and R. Mendes, "Population structure and particle swarm performance," in *Proc. IEEE Congr. Evol. Comput.*, pp. 1671-1676, 2002.

[13] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance," in *Proc. IEEE Congr. Evol. Comput.*, pp. 1931-1938, 1999.

[14] R. A. Krohling, and E. Mendel, "Bare bones particle swarm optimization with Gaussian or Cauchy jumps," in *Proc. IEEE Congr. Evol. Comput.*, pp. 3285-3291, 2009.

[15] J. Kennedy, "Bare bones particle swarms," in *Proc. IEEE SIS*, pp. 80-87, 2003.

[16] J. Li, J. Zhang, C. Jiang, and M. Zhou, "Composite Particle Swarm Optimizer With Historical Memory for Function Optimization," *IEEE Trans. Cybern.,* vol. 45, no. 10, pp. 2350 - 2363, 2015.

[17] Z.-H. Zhan, J. Zhang, Y. Li, and Y.-H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Trans. Evol. Comput.,* vol. 15, no. 6, pp. 832-847, 2011.

[18] Z. Ren, A. Zhang, C. Wen, and Z. Feng, "A scatter learning particle swarm optimization algorithm for multimodal problems," *IEEE Trans. Cybern.,* vol. 44, no. 7, pp. 1127-1140, 2014.

[19] Y.-J. Gong, J.-J. Li, Y. Zhou, Y. Li, H. S.-H. Chung, Y.-H. Shi, and J. Zhang, "Genetic learning particle swarm optimization," *IEEE Trans. Cybern.,* vol. 46, no. 10, pp. 2277 - 2290, 2016.

[20] J. E. Fieldsend, R. M. Everson, and S. Singh, "Using unconstrained elite archives for multiobjective optimization," *IEEE Trans. Evol. Comput.,* vol. 7, no. 3, pp. 305-323, 2003.

[21] S. Yang, "Genetic algorithms with memory-and elitism-based immigrants in dynamic environments," *Evol. Comput.,* vol. 16, no. 3, pp. 385-416, 2008.

[22] S. Xiang, and Y. G. He, "A Quantum-Inspired Evolutionary Algorithm with Elite Group Guided," in *Appl. Mech. Mater.*, pp. 323-333, 2015.

[23] Y. Qin, J. Ji, and C. Liu, "An entropy-based multiobjective evolutionary algorithm with an enhanced elite mechanism," *Appl. Comput. Intell. Soft Comput.,* vol. 2012, pp. 17, 2012.

[24] H. Zhang, D. Yue, X. Xie, S. Hu, and S. Weng, "Multi-elite guide hybrid differential evolution with simulated annealing technique for dynamic economic emission dispatch," *Appl. Soft Comput.,* vol. 34, pp. 312-323, 2015.

[25] Y. Huo, Y. Zhuang, J. Gu, and S. Ni, "Elite-guided multi-objective artificial bee colony algorithm," *Appl. Soft Comput.,* vol. 32, pp. 199-210, 2015.

[26] L. Cui, G. Li, Q. Lin, Z. Du, W. Gao, J. Chen, and N. Lu, "A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation," *Inf. Sci.,* vol. 367, pp. 1012-1044, 2016.

[27] G. Xu, and X. Wang, "Elite-Guiding Binary Differential Evolution," in *ICMSE*, pp. 860-865, 2015.

[28] J. J. Liang, X. B. Mao, B. Y. Qu, B. Niu, and T. J. Chen, "Elite Multi-Group Differential Evolution," in *Proc. IEEE Congr. Evol. Comput.*, pp. 1-7, 2012.

[29] N. Chen, W. N. Chen, Y. J. Gong, Z. H. Zhan, J. Zhang, Y. Li, and Y. S. Tan, "An Evolutionary Algorithm with Double-Level Archives for Multiobjective Optimization," *IEEE Trans. Cybern.,* vol. 45, no. 9, pp. 1851-1863, 2015.

[30] X. Wen, W. N. Chen, Y. Lin, T. Gu, H. Zhang, Y. Li, Y. Yin, and J. Zhang, "A Maximal Clique Based Multiobjective Evolutionary Algorithm for Overlapping Community Detection," *IEEE Trans. Evol. Comput.,* vol. PP, no. 99, pp. 1-15, 2016.

[31] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.,* vol. 6, no. 2, pp. 182-197, 2002.

[32] P. Larrañaga, and J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*: Springer Science & Business Media, 2002.

[33] Q. Yang, W. N. Chen, Y. Li, C. L. P. Chen, X. M. Xu, and J. Zhang, "Multimodal Estimation of Distribution Algorithms," *IEEE Trans. Cybern.,* vol. PP, no. 99, pp. 1-15, 2016.

[34] J. Zhang, and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.,* vol. 13, no. 5, pp. 945-958, 2009.

[35] Q. Yang, W. N. Chen, Z. Yu, T. Gu, Y. Li, H. Zhang, and J. Zhang, "Adaptive Multimodal Continuous Ant Colony Optimization," *IEEE Trans. Evol. Comput.,* vol. PP, no. 99, pp. 1-15, 2016.

[36] M. A. Potter, *The design and analysis of a computational model of cooperative coevolution*: Ph.D. dissertation, George Mason University, 1997.

[37] F. Van den Bergh, and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. Evol. Comput.,* vol. 8, no. 3, pp. 225-239, 2004.

[38] X. Li, and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Trans. Evol. Comput.,* vol. 16, no. 2, pp. 210-224, 2012.

[39] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Trans. Evol. Comput.,* vol. 18, no. 3, pp. 378 - 393, 2014.

[40] Y. Sun, M. Kirley, and S. K. Halgamuge, "Extended Differential Grouping for Large Scale Global Optimization with Direct and Indirect Variable Interactions," in *Proc. Conf. Genet. Evol. Comput.,*, pp. 313-320, 2015.

[41] Y. Mei, M. N. Omidvar, X. Li, and X. Yao, "A Competitive Divide-and-Conquer Algorithm for Unconstrained Large-Scale Black-Box Optimization," *ACM Transactions on Mathematical Software*, in press, 2016.

[42] R. Cheng, and Y. Jin, "A Competitive Swarm Optimizer for Large Scale Optimization," *IEEE Trans. Cybern.,* vol. 45, no. 2, pp. 191-204, 2015.

[43] J. Liang, and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer," in *Proc. IEEE SIS*, pp. 124-129, 2005.

[44] J. J. Liang, and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer with local search," in *Proc. IEEE Congr. Evol. Comput.*, pp. 522-528, 2005.

[45] R. Cheng, C. Sun, and Y. Jin, "A multi-swarm evolutionary framework based on a feedback mechanism," in *Proc. IEEE Congr. Evol. Comput.*, pp. 718-724, 2013.

[46] R. Cheng, and Y. Jin, "A social learning particle swarm optimization algorithm for scalable optimization," *Inf. Sci.,* vol. 291, pp. 43-60, 2015.

[47] K. Tang, X. Li, P. N. Suganthan, Z. Yang, and T. Weise, "Benchmark functions for the CEC 2010 special session and competition on large-scale global optimization," *Technical Report, Nature Inspired Computation and Applications Laboratory, USTC, China*, 2010.

[48] X. Li, K. Tang, M. N. Omidvar, Z. Yang, and K. Qin, "Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization," *Technical Report, Evolutionary Computation and Machine Learning Group, RMIT University, Australia*, 2013.

[49] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Inf. Sci.,* vol. 178, no. 15, pp. 2985-2999, 2008.

[50] Z. Yang, K. Tang, and X. Yao, "Multilevel cooperative coevolution for large scale optimization," in *Proc. IEEE Congr. Evol. Comput.*, pp. 1663-1670, 2008.