

Article

An Entropy-Assisted Particle Swarm Optimizer for Large-Scale Optimization Problem

Weian Guo ^{1,2,*} , Lei Zhu ^{3,*}, Lei Wang ⁴, Qidi Wu ⁴ and Fanrong Kong ⁵¹ Key Laboratory of Intelligent Computing & Signal Processing (Ministry of Education), Anhui University, Hefei 230039, China² Sino-German College of Applied Sciences, Tongji University, Shanghai 201804, China³ Key Lab of Information Network Security Ministry of Public Security, Shanghai 201112, China⁴ School of Electronics and Information Engineering, Tongji University, Shanghai 201804, China; wanglei@tongji.edu.cn (L.W.); wuqidi@tongji.edu.cn (Q.W.)⁵ School of Software Engineering, Tongji University, Shanghai 201804, China; ahshicr@163.com

* Correspondence: guoweian@163.com (W.G.); zhulei_shanghai@163.com or zhulei@stars.org.cn (L.Z.)

Received: 7 April 2019; Accepted: 5 May 2019; Published: 9 May 2019



Abstract: Diversity maintenance is crucial for particle swarm optimizer's (PSO) performance. However, the update mechanism for particles in the conventional PSO is poor in the performance of diversity maintenance, which usually results in a premature convergence or a stagnation of exploration in the searching space. To help particle swarm optimization enhance the ability in diversity maintenance, many works have proposed to adjust the distances among particles. However, such operators will result in a situation where the diversity maintenance and fitness evaluation are conducted in the same distance-based space. Therefore, it also brings a new challenge in trade-off between convergence speed and diversity preserving. In this paper, a novel PSO is proposed that employs competitive strategy and entropy measurement to manage convergence operator and diversity maintenance respectively. The proposed algorithm was applied to the large-scale optimization benchmark suite on CEC 2013 and the results demonstrate the proposed algorithm is feasible and competitive to address large scale optimization problems.

Keywords: diversity maintenance; particle swarm optimizer; entropy; large scale optimization

1. Introduction

Swarm intelligence plays a very active role in optimization areas. As a powerful tool in swarm optimizers, particles swarm optimizer (PSO) has been widely and successfully applied to many different areas, including electronics [1], communication technique [2], energy forecasting [3], job-shop scheduling [4], economic dispatch problems [5], and many others [6]. In the design PSO, each particle has two properties that are velocity and position, respectively. For each generation in the algorithm, particles' properties update according to the mechanisms presented in Equations (1) and (2).

$$\begin{aligned} V_i(t+1) &= \omega V_i(t) + c_1 R_1 (P_{i,pbest}(t) - P_i(t)) \\ &+ c_2 R_2 (P_{gbest}(t) - P_i(t)) \end{aligned} \quad (1)$$

$$P_i(t+1) = P_i(t) + V_i(t+1) \quad (2)$$

where $V_i(t)$ and $P_i(t)$ are used to represent the velocity and position of the i th particle in the t th generation. $\omega \in [0, 1]$ is an inertia weight and $c_1, c_2 \in [0, 1]$ are acceleration coefficients. $R_1, R_2 \in [0, 1]^n$ are two random vectors, where n is the dimension of the problem. $P_{i,pbest}(t)$ is the best position where the i th particle ever arrived, while P_{gbest} is the current global best position found by the whole swarm so far.

According to the update mechanism of PSO, the current global best particle P_{gbest} attracts the whole swarm. However, if P_{gbest} is a local optimal position, it is very difficult for the whole swarm to get rid of it. Therefore, for PSO, it is a notorious problem that the algorithm lacks competitive ability in diversity maintenance, which usually causes a premature convergence or a stagnation in convergence. To overcome this issue, many works are proposed in current decades, which are presented in Section 2 in detail. However, since diversity maintenance and fitness evaluation are conducted in the same distance-based space, it is difficult to distinguish the role of an operator in exploration and exploitation, respectively. It is also a big challenge to explicitly balance the two abilities. Hence, in current research, the proposed methods usually encounter problems, such as structure design, parameter tuning and so on. To overcome the problem, in this paper, on one the hand, we propose a novel method to maintain swarm diversity by an entropy measurement, while, on the other hand, a competitive strategy is employed for swarm convergence. Since entropy is a frequency measurement, while competitive strategy is based on the Euclidean space, the proposed method eliminates the coupling in traditional way to balance exploration and exploitation.

The rest of this paper is organized as follows. In Section 2, the related work to enhance PSO's ability in diversity maintenance is introduced. In Section 3, we propose a novel algorithm named entropy-assisted PSO, which considers convergence and diversity maintenance simultaneously and independently. The experiments on the proposed algorithm are presented in Section 4. We also select several peer algorithms in the comparisons to validate the optimization ability. The conclusions and future works are proposed in Section 5.

2. Related Work

Considering that the standard PSO has the weakness in diversity maintenance, many researchers focused on this topic to improve PSO. By mimicking genetic algorithms, mutation operators are adopted in PSO's design. In [7–9], the authors applied different kinds of mutation operators including Gaussian mutation operator, wavelet mutation operator and so forth to swarm optimizers. In this way, the elements in a particle will be changed according to probabilities and therefore the particle's position changes. However, the change will causes a break down of the convergence process, which is harmful to algorithm's performance. To address the issue, some researchers predefined a threshold to activate mutation operator, which means that mutation operator does not always work, but only happens when the swarm diversity worsens. In [10], a distance-based limit is predefined to activate mutation operator so that the method preserves swarm diversity. A similar idea is adopted in [9], where a Gauss mutation operator is employed. However, as mentioned in [11], for the design of mutation operator, it is difficult to well preset a suitable mutation rate. A large value of mutation rate will result in a loss for the swarm in convergence, while a small value of mutation rate is helpless to preserve swarm diversity.

Besides mutation operator, several other strategies will be activated when the swarm diversity is worse than a predefined limit. Since many distance-indicators, such variance of particles' positions, are employed to evaluate swarm diversity, a natural idea is to increase the distances among particles. In [12], the authors defined a criticality to evaluate the current state of swarm diversity is suitable or not. A crowded swarm has a high value of criticality, while a sparse swarm's criticality is small. A relocating strategy will be activated to disperse the swarm if the criticality is larger than a preset limit. Inspired from the electrostatics, in [13,14], the authors endowed particles with a new property named charging status. For any two charged particles, an electrostatic reaction is launched to regulate their velocities and therefore the charged particles will not be too close. Nevertheless, in the threshold-based design, it is a big challenge to preset the suitable threshold for different optimization problems. In addition, even in one optimization process, the weights of exploitation and exploration are different, it is very difficult to suitably regulate the threshold.

To avoid presetting a threshold, many researchers proposed adaptive way to maintain swarm diversity. The focus is on the parameters setting in PSO's update mechanism. For PSO, there are three components involved in velocity update. The first is inertia component which plays the role to retain

each particle's own property [15,16]. As shown in Equation (1), the value of ω is used to control the weight of this component. A large value of ω helps swarm explore the searching space, while a small value of ω assists a swarm on the exploitation. To help a swarm shift the role from exploration to exploitation, an adaptive strategy is proposed in [15]; by the authors' experience, the value of ω decreasing from 0.9 to 0.4 is helpful for a swarm to properly explore and exploit the searching space. For the cognitive component and social component, which are the second term and third term in Equation (1), they focused more on exploration and exploitation, respectively. To provide an adaptive way to tune their weights, Hu et al constructed several mathematical functions empirically [16,17], which can dynamically regulate the weights of the two components. Besides parameter setting, researchers also provided novel structures for swarm searching. A common way is multi-swarm strategy, which means a whole swarm is divided into several sub-swarms. Each sub-swarm has different roles. On the one hand, to increase the diversity of exemplars, niching strategies are proposed. The particles in the same niche are considered as similar ones, and no information sharing occurs between similar particles. In this way, the searching efficiency is improved. However, the strategy provides a sensitive parameter, e.g. niching radius, in algorithm design. To address this problem, Li used a localized ring topology to propose a parameter-free niching strategy, which improves algorithm design [18]. On the other hand, in multi-swarm strategy, sub-swarms have different tasks. In [19], the authors defined a frequency to switch exploitation and exploration for different sub-swarms, which assists the whole swarm converge and maintain diversity in different optimization phases.

However, in the current research, the diversity measurement and management are conducted in distance-based space, where fitness evaluations are also done. In this way, both particles' quality evaluation and diversity assessment have a heavy coupling. It is very hard to tell the focus on exploitation and exploration of a learning operator. Hence, the algorithms' performances are very sensitive to the design of algorithm's structure and the parameters tuning, which brings a big challenge for users' implementation. To address the issues, in this paper, the contributions are listed as follows. First, we propose a novel way to measure population diversity by entropy, which is from the view of frequency. Second, based on the maximum entropy principle, we propose a novel idea in diversity management. In this way, the exploitation and exploration are conducted independently and simultaneously, which eliminates the coupling the convergence and diversity maintenance and provides a flexible algorithm's structure for users in real implementations.

3. Algorithm

In traditional PSO, both diversity maintenance operator and fitness evaluation operator are conducted in distance-based measurement space. This will result in a heavy coupling in particles' update for exploitation and exploration, which brings a big challenge in balance the weights of the two abilities. To overcome this problem, in this paper, we propose a novel idea to improve PSO which is termed entropy-assisted particle swarm optimization (EAPSO). In the proposed algorithm, we consider both diversity maintenance and fitness evaluation independently and simultaneously. Diversity maintenance and fitness evaluation are conducted by frequency-based space and distance-based space, respectively. To reduce the computation load in large scale optimization problem, in this paper, we only consider the phenotypic diversity, which is depicted by fitness domain, rather than genetic diversity. In each generation, the fitness domain is divided into several segments. We account the number of particles in each segment, as shown in Figure 1.

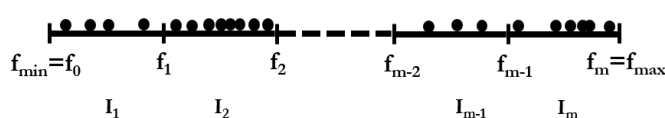


Figure 1. The illustration for the entropy diversity measurement.

The maximum fitness and minimal fitness are set as the fitness landscape boundaries. For the landscape, it is uniformly divided into several segments. For each segment, we account for the number of particles, namely number of fitness values. For the entropy calculation, we use the following formulas, which are inspired by Shannon entropy.

$$H = - \sum_i^m p_i \log p_i \quad (3)$$

where H is used to depict the entropy of a swarm and p_i is the probability that fitness values are located in the i th segment, which can be obtained by Equation (4).

$$p_i = \frac{num_i}{m} \quad (4)$$

where m is the swarm size, and num_i is the number of fitness values that appear in the i th segment. Inspired by the maximum entropy principle, the value of H is maximized if and only if $p_i = p_j$, where $i, j \in [1, n]$. Hence, to gain a large value of entropy, the fitness values are supposed to be distributed uniformly in all segments. To pursue this goal, we define a novel measurement to select global best particle, which considers fitness and entropy simultaneously. All particles are evaluated by Equation (5).

$$Q_i = \alpha fitness_{rank} + \beta entropy_{rank} \quad (5)$$

where $fitness_{rank}$ is the fitness value rank of a particle, while $entropy_{rank}$ is the entropy value of a particle. α and β are employed to manage the weights of the two ranks. However, in real applications, the tuning on two parameters will increase the difficulty. Considering that the two parameters are used to adjust the weights of exploration and exploitation respectively, in real applications, we fix the value of one of them, while tuning the other one. In this paper, we set the weight of β as 1, and therefore, by regulating the value of α , the weights of exploration and exploitation can be adjusted. To calculate the value of $fitness_{rank}$, all particles are ranked according to their fitness values. For a particle's $entropy_{rank}$, it is defined as the segment rank where the particle is. A segment has a top rank if there is few particles in the segment, while a segment ranks behind if there are crowded particles in the segment. According to Equation (5), a small value of Q_i means a good performance of particle i .

In the proposed algorithm, we propose a novel learning mechanism as shown in Equation (6). We randomly and uniformly divide a swarm into several groups. Namely, the numbers of particles in each group are equal. The particle with high quality of Q in a group is considered as an exemplar, which means that the exemplars are selected according to both fitness evaluation and entropy selection. In this paper, we abandon the historical information, but only use the information of the current swarm, which reduces the spacing complexity of the algorithm. The update mechanism of the proposed algorithm is given in Equation (6).

$$\begin{aligned} V_i(t+1) &= \omega V_i(t) \\ &+ r_1 * c_1 * (P_{lw}(t) - P_{ll}(t)) \\ &+ r_2 * c_2 * (P_g - P_{ll}(t)) \end{aligned} \quad (6)$$

$$P_{ll}(t+1) = P_{ll}(t) + V_i(t+1) \quad (7)$$

where V_i is the velocity of the i th particle, ω is the inertia parameter, P_{lw} is the position of the local winner in a group, P_{ll} is the position of local loser in the same group, P_g is the current best position found, c_1 and c_2 are the cognitive coefficient and social coefficient respectively, r_1 and r_2 are random values that belong to $[0, 1]$, and t is used to present the index of generation. On the one hand, the fitness is evaluated according to the objective function. On the other hand, the divergence situation of a particle is evaluated by the entropy measurement. By assigning weights to the divergence and

convergence, the update mechanism involves both exploration and exploitation. The pseudo-codes of the proposed algorithms are given in Algorithm 1.

Algorithm 1: Pseudo-codes of entropy-assisted PSO.

Input: Swarm size n , Group size g , Number of segments m , Weight value α

Output: The current global best particle

- 1 Loop 1: Evaluate the fitness for all particles, and for the i th particle, its fitness is f_i ;
 - 2 Set the maximum value and minimal value of fitness as f_{max} and f_{min} respectively;
 - 3 Divide the interval $[f_{min}, f_{max}]$ into m segments;
 - 4 Calculate the number of fitness values in each segment, for the i th segment, the number of fitness values is recorded as num_i ;
 - 5 Sort the number of fitness values, and record the fitness rank fr_i for each particle;
 - 6 Sort the segments according to the number of fitness values, and record the segment rank sr_i for each particle;
 - 7 Evaluate each particle' quality Q by Equation (5);
 - 8 Divide the swarm into g groups, and compare the particles by their performances Q ;
 - 9 Select the global best particle according to Q ;
 - 10 Update particles according to Equation (6);
 - 11 If the termination condition is satisfied, output the current global best particle; Otherwise, goto Loop 1;
-

In the proposed algorithm, for each particle, it has two exemplars, which are local best exemplar and global best particle respectively. The ability to maintain diversity is improved from two aspects. First, in the evaluation of particles, we consider both fitness and diversity by objective function and entropy respectively. Second, we divide a swarm into several sub-swarms, the number of local best exemplars are equal to the number of sub-swarms. In this way, even some exemplars are located in local optimal positions, and they will not affect the whole swarm so that the diversity of exemplars is maintained. Finally, in this paper, the value of α in Equation (5) provides an explicit way to manage the weights of exploration ability and exploitation ability and therefore eliminate the coupling of the two abilities.

4. Experiments and Discussions

We applied the proposed algorithm to large scale optimization problems (LSOPs). In general, LSOPs have hundreds or thousands of variables. For meta-heuristic algorithms, they usually suffer from “the curse of dimensionality”, which means the performance deteriorates dramatically as the problem dimension increases [20]. Due to a large number of variables, the searching space is complex, which brings challenges for meta-heuristic algorithms. First, the searching space is huge and wide, which demands of a high searching efficiency [21,22]. Second, the large scale causes capacious attraction domains of local optimum and exacerbates the difficulty for algorithms to get rid of local optimal positions [23]. Hence, in the optimization process, both convergence ability and diversity maintenance of a swarm are crucial to algorithm's performance. We employed LSOPs in CEC 2013 as benchmark suits to test the proposed algorithm. The details of the benchmarks are listed in [24]. In comparisons, several peer algorithms, including DECC-dg (Cooperative Co-Evolution with Differential Grouping), MMOCC (Multimodal optimization enhanced cooperative coevolutio), SLPSO (Social Learning Particle Swarm Optimization), and CSO (Competitive Swarm Optimizer), were selected. DECC-dg is an improved version of DECC, which is reported in [25]. CSO was proposed by Cheng and Jin, which exhibits a powerful ability in dealing with the large scale optimization problems of IEEE CEC 2008 [21]. SLPSO was proposed by the same authors in [22], where a social learning concept is employed. For MMOCC, which is currently proposed by Peng etc, which adopts the idea of CC framework and the techniques of multi-modular optimization [26].

For each algorithm, we present a mean performance of 25 independent runs. The termination condition was limited by the maximum number of Fitness Evaluations (FEs), i.e., 3×10^6 , as recommended in [24]. For EA-PSO, the population size was 1200. The reasons to employ a large size of population are presented as follows: First, a large size of population enhances the parallel computation ability for the algorithm. Second, the grouping strategy will be more efficient when a large size of population is employed. If the population size is too small, the size of groups will also be small and therefore the learning efficiency in each group decreases. Third, in EA-PSO, the diversity management is conducted by entropy control, which is a frequency based approach. As mentioned in [27], a large population size is recommended when using FBMs. Fourth, a large population size is helpful to avoid empty segments. Although a large size of population was employed, we used the number of fitness evaluations (FEs) to limit the computational resources in comparisons to guarantee a fair comparison. The number of intervals (m) was 30. The group size and α were set as 20 and 0.1, respectively. The experimental results are presented in Table 1. The best results of mean performance for each benchmark function are marked by bold font. To provide a statistical analysis, the p values were obtained by Wilcoxon signed ranks test. According to the statistical analysis, most of the p values were smaller than 0.05, which demonstrates the differences were effective. However, for benchmark F6, the comparisons “EA-PSO vs. CSO” and “EA-PSO vs. DECC-DG”, the p values were larger than 0.05, which means that there was no significant differences between the algorithms’ performance for the benchmark. The same was found for “EA-PSO vs. MMO-CC” on benchmark F8, “EA-PSO vs. SLPSO” on benchmark F12.

According to Table 1, EA-PSO outperformed the other algorithms for 10 benchmark functions. For F2, F4, F12, and F13, EA-PSO took the second or third ranking in the comparisons. The comparison results demonstrate that the proposed algorithm is very competitive to address large scale optimization problems. We present the convergence profiles of different algorithms in Figure 2.

Table 1. The experimental results of 1000-dimensional IEEE CEC’ 2013 benchmark functions with fitness evaluations of 3×10^6 . The best performance is marked by bold font in each line.

Function	Quality	CSO	SLPSO	DECC-DG	MMO-CC	EA-PSO
F_1	mean	3.68×10^{-17}	3.70×10^{-14}	2.79×10^6	4.82×10^{-20}	3.53×10^{-16}
	std	3.70×10^{-19}	1.44×10^{-15}	6.70×10^5	1.30×10^{-21}	1.14×10^4
	p -value	1.22×10^{-18}	2.31×10^{-18}	3.40×10^{-4}	9.76×10^{-21}	-
F_2	mean	7.08×10^2	6.70×10^3	1.41×10^4	1.51×10^3	1.45×10^3
	std	7.04×10^0	4.98×10^1	3.03×10^2	8.43×10^0	4.21×10^1
	p -value	1.57×10^{-5}	9.70×10^{-27}	9.57×10^{-27}	4.85×10^{-24}	-
F_3	mean	2.16×10^1	2.16×10^1	2.07×10^1	2.06×10^1	2.15×10^1
	std	1.39×10^{-3}	1.14×10^{-3}	2.19×10^{-3}	2.36×10^{-3}	2.26×10^{-3}
	p -value	1.41×10^{-5}	2.01×10^{-2}	1.44×10^{-36}	1.53×10^{-36}	-
F_4	mean	1.14×10^{10}	1.20×10^{10}	6.72×10^{10}	5.15×10^{11}	4.36×10^9
	std	2.66×10^8	5.54×10^8	5.76×10^9	9.71×10^{10}	7.72×10^8
	p -value	7.92×10^{-12}	4.94×10^{-12}	6.55×10^{-11}	2.57×10^{-11}	-
F_5	mean	7.44×10^5	7.58×10^5	3.13×10^6	2.42×10^6	6.68×10^5
	std	2.49×10^4	2.14×10^4	1.23×10^5	1.14×10^5	3.57×10^5
	p -value	7.57E-09	7.43E-09	3.92×10^{-15}	5.48×10^{-15}	-
F_6	mean	1.06×10^6	1.06×10^6	1.06×10^6	1.06×10^6	1.05×10^6
	std	1.90×10^2	1.64×10^2	3.70×10^2	6.41×10^2	4.91×10^2
	p -value	1.71×10^{-1}	7.06×10^{-3}	3.18×10^{-1}	2.70×10^{-3}	-
F_7	mean	8.19×10^6	1.73×10^7	3.45×10^8	1.28×10^{10}	1.43×10^6
	std	4.85×10^5	1.49×10^6	7.60×10^7	1.07×10^9	4.87×10^6
	p -value	7.06×10^{-14}	3.18×10^{-11}	2.76×10^{-4}	4.61×10^{-12}	-
F_8	mean	3.14×10^{14}	2.89×10^{14}	1.73×10^{15}	1.54×10^{14}	1.47×10^{14}
	std	1.09×10^{13}	1.75×10^{13}	2.78×10^{14}	4.45×10^{13}	6.17×10^{12}
	p -value	9.71×10^{-15}	8.23×10^{-11}	3.17×10^{-6}	9.50×10^{-1}	-
F_9	mean	4.42×10^7	4.44×10^7	2.79×10^8	1.76×10^8	5.05×10^7
	std	1.59×10^6	1.47×10^6	1.32×10^7	7.03×10^6	1.17×10^7
	p -value	4.38×10^{-6}	3.81×10^{-6}	7.65×10^{-12}	1.86×10^{-12}	-
F_{10}	mean	9.40×10^7	9.43×10^7	9.43×10^7	9.38×10^7	9.35×10^7
	std	4.28×10^4	3.99×10^4	6.45×10^4	1.02×10^5	7.92×10^4
	p -value	4.89×10^{-1}	3.81×10^{-4}	7.65×10^{-3}	1.86×10^{-5}	-

Table 1. Cont.

Function	Quality	CSO	SLPSO	DECC-DG	MMO-CC	EA-PSO
F_{11}	mean	3.56×10^8	9.98×10^9	1.26×10^{11}	5.66×10^{12}	5.00×10^8
	std	1.47×10^7	1.82×10^9	2.44×10^{10}	1.09×10^{12}	1.92×10^7
	p-value	6.46×10^{-15}	7.09×10^{-5}	7.54×10^{-5}	2.05×10^{-5}	-
F_{12}	mean	1.39×10^3	1.13×10^3	5.89×10^7	1.14×10^{11}	1.40×10^3
	std	2.19×10^1	2.12×10^1	2.75×10^6	6.32×10^{10}	2.23×10^1
	p-value	2.76×10^{-10}	6.79×10^{-1}	6.55×10^{-17}	1.62×10^{-3}	-
F_{13}	mean	1.75×10^9	2.05×10^9	1.06×10^{10}	1.32×10^{12}	1.66×10^9
	std	6.47×10^7	2.13×10^8	7.94×10^8	2.88×10^{11}	5.54×10^7
	p-value	1.19×10^{-11}	4.98×10^{-9}	5.97×10^{-12}	5.85×10^{-15}	-
F_{14}	mean	6.95×10^9	1.60×10^{10}	3.69×10^{10}	4.12×10^{11}	1.40×10^8
	std	9.22×10^8	1.62×10^9	6.58×10^9	1.21×10^{11}	2.79×10^7
	p-value	7.51×10^{-7}	2.55×10^{-10}	5.05×10^{-5}	6.99×10^{-5}	-
F_{15}	mean	1.65×10^7	6.68×10^7	6.32×10^6	4.05×10^8	7.69×10^6
	std	2.21×10^5	1.01×10^6	2.69×10^5	1.91×10^7	3.39×10^5
	p-value	8.91×10^{-23}	5.47×10^{-25}	1.49×10^{-25}	2.57×10^{-4}	-

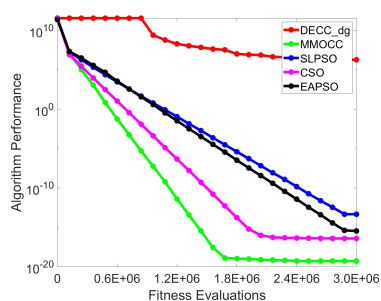
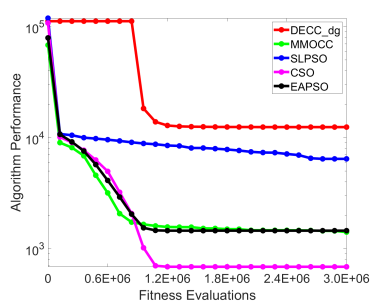
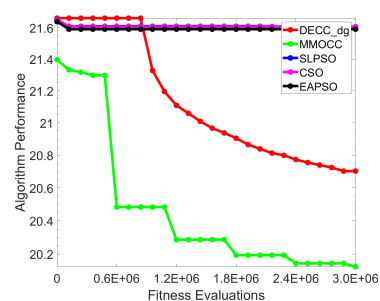
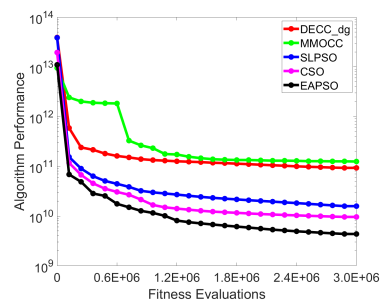
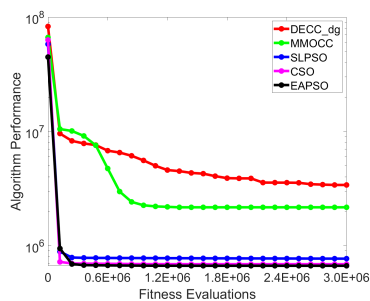
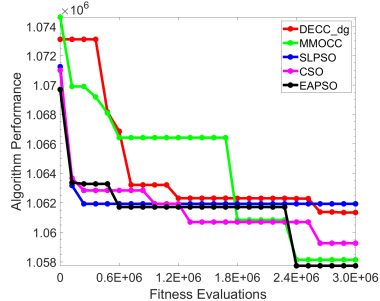
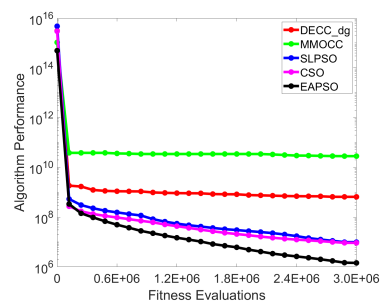
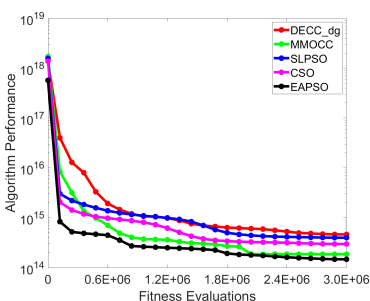
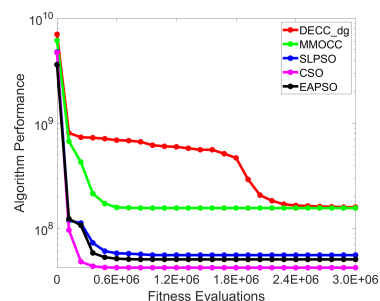
(a) F_1 (b) F_2 (c) F_3 (d) F_4 (e) F_5 (f) F_6 (g) F_7 (h) F_8 (i) F_9

Figure 2. Cont.

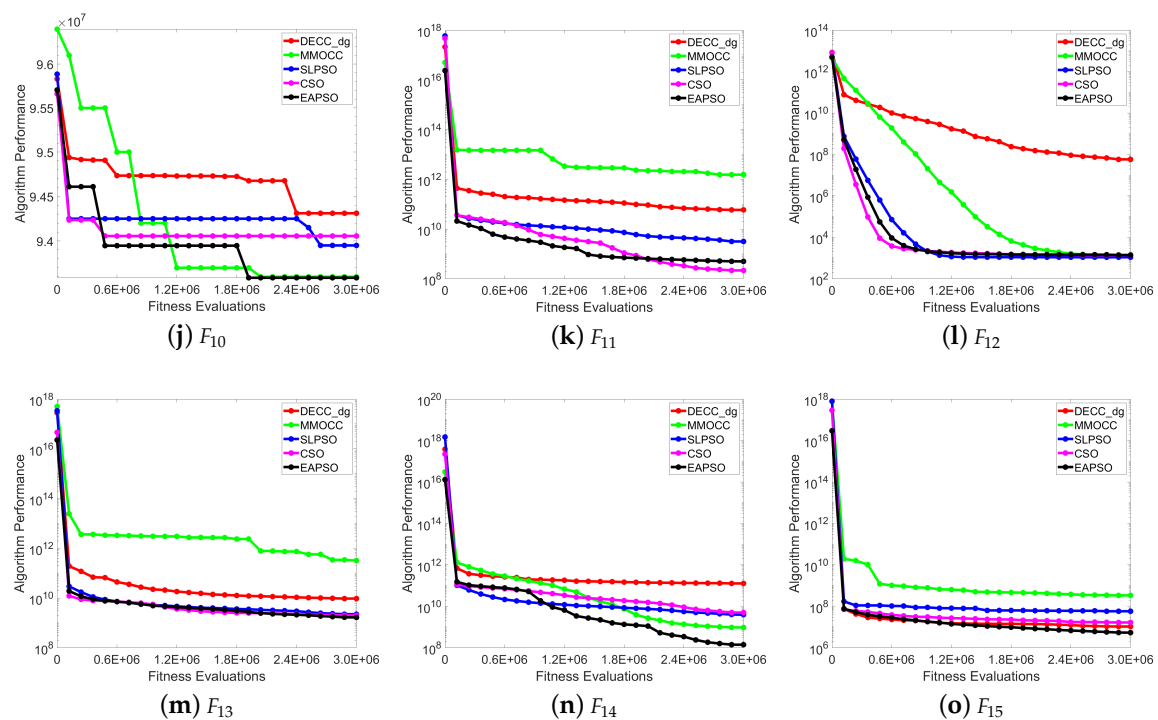


Figure 2. Convergence profiles of different algorithms obtained on the CEC'2013 test suite with 1000 dimensions.

In this study, the value of α was used to balance the abilities of exploration and exploitation. Hence, we investigated the influence of α to algorithm's performance. In this test, we set α as 0.2, 0.3 and 0.4. For other parameters, we used the same setting as in Table 1. For each value of α , we ran the algorithm 25 times and present the mean optimization results in Table 2. According to the results, there was no significant difference in the order of magnitude. On the other hand, for the four values, $\alpha = 0.1$ and $\alpha = 0.2$, both won six times, which demonstrates that a small value of α would help the algorithm achieve a more competitive optimization performance. The convergence profiles for algorithm's performances with different values of α are presented in Figure 3.

Table 2. The different values of α to EA-PSO's performances on IEEE CEC 2013 large scale optimization problems with 1000 dimensions (fitness evaluations = 3×10^6).

Function	$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$
F1	3.53×10^{-16}	2.97×10^{-16}	5.07×10^{-16}	9.43×10^{-16}
F2	1.45×10^3	1.45×10^3	1.58×10^3	1.45×10^3
F3	2.15×10^1	2.15×10^1	2.15×10^1	2.15×10^1
F4	4.36×10^9	6.37×10^9	6.97×10^9	9.02×10^9
F5	6.68×10^5	5.48×10^5	8.72×10^5	6.87×10^5
F6	1.06×10^6	1.06×10^6	1.06×10^6	1.06×10^6
F7	1.43×10^6	2.02×10^6	2.51×10^7	9.86×10^6
F8	1.47×10^{14}	3.11×10^{13}	1.29×10^{14}	8.66×10^{13}
F9	5.05×10^7	4.59×10^7	5.79×10^7	7.02×10^7
F10	9.35×10^7	9.40×10^7	9.41×10^7	9.42×10^7
F11	5.00×10^8	4.98×10^8	3.74×10^8	4.23×10^8
F12	1.40×10^3	1.30×10^3	1.33×10^3	1.51×10^3
F13	1.66×10^9	7.38×10^8	1.61×10^9	5.86×10^8
F14	1.40×10^8	1.44×10^8	4.21×10^8	4.87×10^8
F15	7.69×10^6	7.42×10^6	8.04×10^6	7.65×10^6

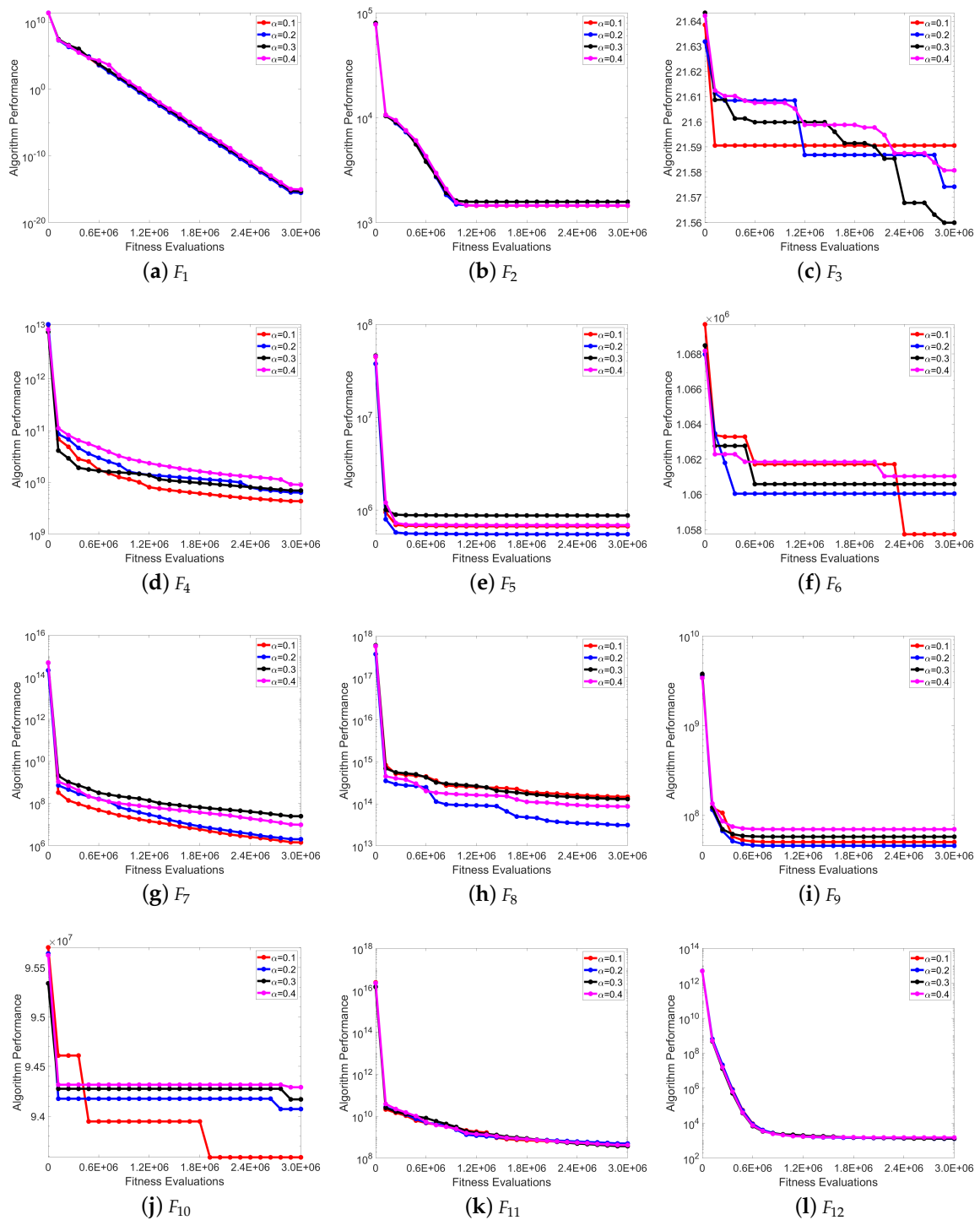


Figure 3. Cont.

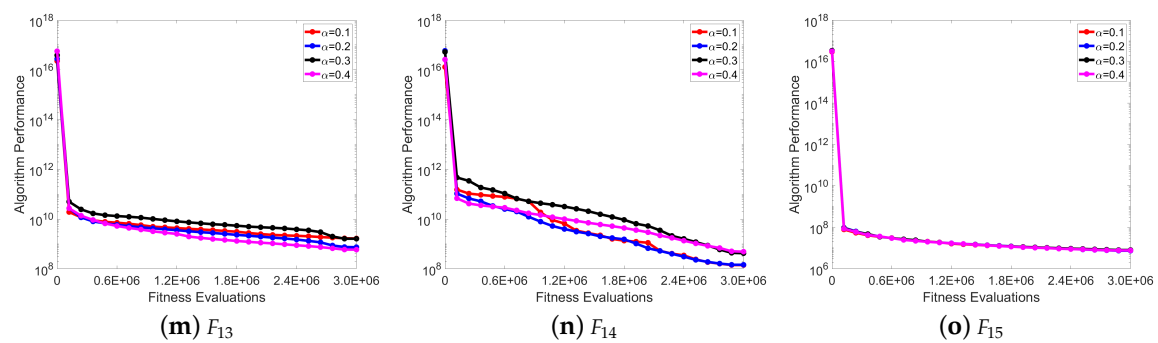


Figure 3. Convergence profiles of different algorithms obtained on the CEC'2013 test suite with 1000 dimensions.

5. Conclusions

In this paper, a novel particle swarm optimizer named entropy-assisted PSO is proposed. All particles are evaluated by fitness and diversity simultaneously and the historical information of the particles are no longer needed in particle update. The optimization experiments were conducted on the benchmarks suite of CEC 2013 with the topic of large scale optimization problems. The comparison results demonstrate the proposed structure helped enhance the ability of PSO in addressing large scale optimization and the proposed algorithm EA-PSO achieved competitive performance in the comparisons. Moreover, since the exploration and exploitation are conducted independently and simultaneously in the proposed structure, the proposed algorithm's structure is flexible to many different kinds of optimization problems.

In the future, the mathematical mechanism of the proposed algorithm will be further investigated and discussed. Considering that, for many other kinds of optimization problems, such as multi-modal optimization problems, dynamic optimization problems, and multi-objective optimization, the population divergence is also crucial to algorithms' performances, we will apply the entropy idea to such problems and investigate the roles in divergence maintenance.

Author Contributions: Conceptualization, W.G., L.W. and Q.W.; Methodology, W.G.; Software, W.G. and L.Z.; Validation, W.G. and L.Z.; Formal analysis, W.G. and L.W.; Investigation, W.G. and F.K.; Resources, W.G. and L.Z.; Data curation, W.G.; Writing original draft preparation, W.G. and F.K.; Writing review and editing, W.G., L.W. and Q.W.; Visualization, W.G. and L.Z.; Supervision, L.W. and Q.W.; Project administration, L.W. and Q.W.; Funding acquisition, W.G. and L.Z.

Funding: This work was sponsored by the National Natural Science Foundation of China under Grant Nos. 71771176 and 61503287, and supported by Key Lab of Information Network Security, Ministry of Public Security and Key Laboratory of Intelligent Computing & Signal Processing, Ministry of Education.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Shi, H.; Wen, H.; Hu, Y.; Jiang, L. Reactive Power Minimization in Bidirectional DC-DC Converters Using a Unified-Phasor-Based Particle Swarm Optimization. *IEEE Trans. Power Electron.* **2018**, *33*, 10990–11006. [\[CrossRef\]](#)
- Bera, R.; Mandal, D.; Kar, R.; Ghoshal, S.P. Non-uniform single-ring antenna array design using wavelet mutation based novel particle swarm optimization technique. *Comput. Electr. Eng.* **2017**, *61*, 151–172. [\[CrossRef\]](#)
- Osorio, G.J.; Matias, J.C.O.; Catalao, J.P.S. Short-term wind power forecasting using adaptive neuro-fuzzy inference system combined with evolutionary particle swarm optimization, wavelet transform and mutual information. *Renew. Energy* **2015**, *75*, 301–307. [\[CrossRef\]](#)

4. Nouiri, M.; Bekrar, A.; Jemai, A.; Niar, S.; Ammari, A.C. An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem. *J. Intell. Manuf.* **2018**, *29*, 603–615. [[CrossRef](#)]
5. Aliyari, H.; Effatnejad, R.; Izadi, M.; Hosseini, S.H. Economic Dispatch with Particle Swarm Optimization for Large Scale System with Non-smooth Cost Functions Combine with Genetic Algorithm. *J. Appl. Sci. Eng.* **2017**, *20*, 141–148. [[CrossRef](#)]
6. Bonyadi, M.R.; Michalewicz, Z. Particle swarm optimization for single objective continuous space problems: A review. *Evol. Comput.* **2017**, *25*, 1–54. [[CrossRef](#)]
7. Higashi, N.; Iba, H. Particle swarm optimization with gaussian mutation. In Proceedings of the 2003 IEEE Swarm Intelligence Symposium, Indianapolis, IN, USA, 26 April 2003; pp. 72–79.
8. Ling, S.H.; Iu, H.H.C.; Chan, K.Y.; Lam, H.K.; Yeung, B.C.W.; Leung, F.H. Hybrid particle swarm optimization with wavelet mutation and its industrial applications. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2008**, *38*, 743–763. [[CrossRef](#)]
9. Wang, H.; Sun, H.; Li, C.; Rahnamayan, S.; Pan, J.-S. Diversity enhanced particle swarm optimization with neighborhood search. *Inf. Sci.* **2013**, *223*, 119–135. [[CrossRef](#)]
10. Sun, J.; Xu, W.; Fang, W. A diversity guided quantum behaved particle swarm optimization algorithm. In *Simulated Evolution and Learning*; Wang, T.D., Li, X., Chen, S.H., Wang, X., Abbass, H., Iba, H., Chen, G., Yao, X., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4247, pp. 497–504.
11. Jin, Y.; Branke, J. Evolutionary optimization in uncertain environments—A survey. *IEEE Trans. Evol. Comput.* **2005**, *9*, 303–317. [[CrossRef](#)]
12. Lovbjerg, M.; Krink, T. Extending particle swarm optimisers with self-organized criticality. In Proceedings of the 2002 Congress on Evolutionary Computation, Honolulu, HI, USA, 12–17 May 2002; Volume 2, pp. 1588–1593.
13. Blackwell, T.M.; Bentley, P.J. Dynamic search with charged swarms. In Proceedings of the Genetic and Evolutionary Computation Conference, New York, NY, USA, 9–13 July 2002; pp. 19–26.
14. Blackwell, T. Particle swarms and population diversity. *Soft Comput.* **2005**, *9*, 793–802. [[CrossRef](#)]
15. Zhan, Z.; Zhang, J.; Li, Y.; Chung, H.S. Adaptive particle swarm optimization. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2009**, *39*, 1362–1381. [[CrossRef](#)] [[PubMed](#)]
16. Hu, M.; Wu, T.; Weir, J.D. An adaptive particle swarm optimization with multiple adaptive methods. *IEEE Trans. Evol. Comput.* **2013**, *17*, 705–720. [[CrossRef](#)]
17. Liang, J.J.; Qin, A.K.; Suganthan, P.N.; Baskar, S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. Evol. Comput.* **2006**, *10*, 281–295. [[CrossRef](#)]
18. Li, X. Niching without niching parameters: Particle swarm optimization using a ring topology. *IEEE Trans. Evol. Comput.* **2010**, *14*, 150–169. [[CrossRef](#)]
19. Siarry, P.; Pétrowski, A.; Bessaou, M. A multipopulation genetic algorithm aimed at multimodal optimization. *Adv. Eng. Softw.* **2002**, *33*, 207–213. [[CrossRef](#)]
20. Yang, Z.; Tanga, K.; Yao, X. Large scale evolutionary optimization using cooperative coevolution. *Inf. Sci.* **2008**, *178*, 2985–2999. [[CrossRef](#)]
21. Cheng, R.; Jin, Y. A competitive swarm optimizer for large scale optimization. *IEEE Trans. Cybern.* **2015**, *45*, 191–204. [[CrossRef](#)]
22. Cheng, R.; Jin, Y. A social learning particle swarm optimization algorithm for scalable optimization. *Inf. Sci.* **2015**, *291*, 43–60. [[CrossRef](#)]
23. Yang, Q.; Chen, W.-N.; Deng, J.D.; Li, Y.; Gu, T.; Zhang, J. A Level-Based Learning Swarm Optimizer for Large-Scale Optimization. *IEEE Trans. Evol. Comput.* **2018**, *22*, 578–594. [[CrossRef](#)]
24. Li, X.; Tang, K.; Omidvar, M.N.; Yang, Z.; Qin, K. *Benchmark Functions for the CEC 2013 Special Session And Competition on Large-Scale Global Optimization*; Tech. Rep.; School of Computer Science and Information Technology, RMIT University: Melbourne, Australia, 2013.
25. Omidvar, M.N.; Li, X.; Mei, Y.; Yao, X. Cooperative Co-Evolution with Differential Grouping for Large Scale Optimization. *IEEE Trans. Evol. Comput.* **2014**, *18*, 378–393. [[CrossRef](#)]

26. Peng, X.; Jin, Y.; Wang, H. Multimodal optimization enhanced cooperative coevolution for large-scale optimization. *IEEE Trans. Cybern.* **2018**, [[CrossRef](#)]
27. Corriveau, G.; Guilbault, R.; Tahan, A.; Sabourin, R. Review and Study of Genotypic Diversity Measures for Real-Coded Representations. *IEEE Trans. Evol. Comput.* **2012**, *16*, 695–710. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).