

# Comparisons Study of APSO OLPSO and CLPSO on CEC2005 and CEC2014 Test Suits

Yan-Fei Li, *Student Member, IEEE*, Zhi-Hui Zhan (Corresponding Author), *Member, IEEE*, Ying Lin, *Member, IEEE*, and Jun Zhang, *Senior Member, IEEE*

**Abstract**—Particle swarm optimization (PSO) is originally designed to solve continuous optimization problems. Recently, lots of improved PSO variants with different features have been proposed, such as Adaptive particle swarm optimization (APSO), Orthogonal Learning particle swarm optimization (OLPSO) and Comprehensive Learning particle swarm optimization (CLPSO). In order to find out whether these PSOs have any particular difficulties or preference and whether one of them would outperform the others on a majority of the tested problems, we analyze the performance of different PSOs on various tested problems. In this paper, we evaluate the performance of APSO, OLPSO, and CLPSO on more complex benchmark functions. The comparison is performed on a large amount of real-parameter optimization problems, including the CEC 2005 and the CEC 2014 benchmark functions. Finally, we find out that the OLPSO achieves higher solution quality than the other two PSOs on most problems based on the simulation results on benchmark functions.

**Keywords**—Adaptive particle swarm optimization (APSO); Orthogonal Learning particle swarm optimization (OLPSO); Comprehensive Learning particle swarm optimization (CLPSO); benchmark problems

## I. INTRODUCTION

PARTICLE swarm optimization (PSO), which was introduced by Kennedy and Eberhart [1], is one of the most important swarm intelligence paradigms in

Y.-F. Li, Z.-H. Zhan, and J. Zhang are with the Department of Computer Science, Sun Yat-Sen University, Guangzhou, 510275, China, with the School of Advanced Computing, Sun Yat-Sen University, Guangzhou, 510275, China, with the Key Laboratory of Machine Intelligence and Advanced Computing (Sun Yat-sen University), Ministry of Education, China, with the Engineering Research Center of Supercomputing Engineering Software (Sun Yat-sen University), Ministry of Education, China, with the Key Laboratory of Software Technology, Education Department of Guangdong Province, China, and also with State Key Laboratory of Mathematical Engineering and Advanced Computing. Zhi-Hui Zhan is the corresponding author, email: [zhanzh@mail.sysu.edu.cn](mailto:zhanzh@mail.sysu.edu.cn).

Y. Lin is with the Department of Psychology, Sun Yat-Sen University, Guangzhou, 510275, China.

This work was supported in part by the, in part by the National Natural Science Foundations of China (NSFC) with No. 61402545 and No. 61309003, the Natural Science Foundations of Guangdong Province for Distinguished Young Scholars with No. 2014A030306038, the Project for Pearl River New Star in Science and Technology, Guangzhou, China, the Fundamental Research Funds for the Central Universities, the NSFC Key Program with No. 61332002, the NSFC for Distinguished Young Scholars with No. 61125205, the National High-Technology Research and Development Program (863 Program) of China No.2013AA01A212, and the Open Project Program of the State Key Lab of Mathematical Engineering and Advanced Computing.

evolutionary computation community and has attracted more and more attention in recent years [2][3][4]. PSO was designed to imitate the interaction of the bird flocking and fish schooling, it works by utilizing a population of candidate solutions (called particles) searching in the search-space. The particles are guided by both the historical best position of their own (pbest) and the position of the best particle in the swarm (gbest). Because PSO is simple and as well as practical, researchers have found many applications of PSO in optimization problems, due to its easy implement [5][6][7].

However, as PSO is a population-based heuristic stochastic optimization algorithm, it may be time-consuming for obtaining satisfactory results. Moreover, when solving complex multimodal problems, the standard PSO algorithm may easily get trapped in local optima. Hence, accelerating convergence speed and avoiding the local optima have always been the two appealing goals with the development of the PSO [8]. Recently a large amount of efforts have been devoted to the improvement of PSO, and lots of improved PSO variants with different features have been proposed. Mendes et al. [9] proposed a Fully Informed Particle Swarm (FIPS) so as to solve single global optimization problems. In FIPS, all neighbors influence the fly of the particles. Integrated with five types of topology, complete graph, ring, four clusters, pyramid and square, FIPS is able to find the single global peak rapidly. In order to solve multimodal optimization problems, Qu et al. [10] proposed a distance-based Locally Informed Particle Swarm algorithm (LIPS), where each particle is informed by several of its nearest neighbors. Unlike FIPS, LIPS defines a particle's neighbor using Euclidean distance. The experimental results in [10] verify the superiority and consistence of LIPS in multimodal optimization problems. To avoid the quick convergence of standard PSO, A. Muñoz et al. [11] introduced the singly-linked neighborhood structure and the perturbation of the particle's best in their proposed Constrained Optimization via PSO (COPSO) algorithm. In [12], A. Ochoa et al. proposed a model by combining data mining and evolutionary computation to analyze the voting and predict the final ranking of the Eurovision Song Contest, and simulations showed that COPSO contributes competitive results to the prediction performance. Liang et al. [13] proposed the comprehensive learning PSO (CLPSO) to increase its ability to avoid local optima by using all other particles' historical best information to update a particle's velocity. Zhan et al. [8] developed a systematic parameter adaptation scheme by devising an elitist learning strategy

(ELS) and an evolutionary state estimation (ESE) technique to control parameters in their proposed Adaptive PSO (APSO). Moreover, Zhan et al. [14] introduces an orthogonal learning (OL) strategy for PSO to discover more useful information that lies in the particle's own historical best experience and its neighborhood's best experience via orthogonal experimental design, which comes up with the algorithm Orthogonal Learning PSO (OLPSO).

Results show that these PSO algorithms have enhanced the performance of the PSO paradigm in terms of convergence speed, global optimality, solution accuracy, and algorithm reliability on conventional benchmark functions. Nevertheless, according to the theorem of "no free lunch" [15], one algorithm cannot fit every kind of problems perfectly. One algorithm may outperform the other one on unimodal functions, but fail on multimodal functions. In order to investigate the performance of CLPSO, APSO, and OLPSO on more complex benchmark problems, the three PSOs are used to solve the problems within the CEC 2005 and CEC 2014 test suits. The reason for choosing these three algorithms is because they have quite different searching behaviors. This paper aims to analyze different improved PSOs' performance on a large number of variant complex benchmark problems, and find out whether they have any particular difficulties or preference and whether one of them would outperform the others on a majority of the tested problems. The comparisons and discussion of these PSOs' performance are presented in this paper.

Overall, experimental evaluations show that all the tested algorithms yield potential characteristics on different problems, and the OLPSO performs best on most tested functions and reveals more robust compared to APSO and CLPSO.

The rest of the paper is organized as follows. Section II introduces the three PSOs used in the study: APSO, OLPSO, and CLPSO. The experimental results of the CEC 2005 and the CEC 2014 are presented in Section III and Section IV respectively. Finally, conclusions are given in Section V.

## II. BACKGROUND

### A. PSO

PSO uses a swarm of particles to represent the potential solutions of the problem. Each particle  $i$  has a position vector  $X_i=[x_{i1}, x_{i2}, \dots, x_{iD}]$ , a velocity vector  $V_i=[v_{i1}, v_{i2}, \dots, v_{iD}]$ , and a personal historical best position vector  $P_i=[p_{i1}, p_{i2}, \dots, p_{iD}]$ , where  $D$  is the dimensions of the solution space. In the initialization, the position and vector are randomly set within the search space and maximal velocity space respectively. The personal historical best position vector  $P_i$  is set to  $X_i$ . By calculating the fitness values of all the particles, the best position vector of all the particles in the  $i$ th particle's neighborhood is denoted as  $P_n=[p_{n1}, p_{n2}, \dots, p_{nD}]$ .

During the evolutionary process, the velocity  $v_{id}$  and the position  $x_{id}$  of the  $d$ th dimension of the  $i$ th particle are updated as follows:

$$v_{id} = \omega \times v_{id} + c_1 \times r_{1d} \times (p_{id} - x_{id}) + c_2 \times r_{2d} \times (p_{nd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

where  $w$  is the inertia weight,  $c_1$  and  $c_2$  are the acceleration coefficients, and  $r_{1d}$  and  $r_{2d}$  are the two independent random numbers uniformly distributed in the range of  $[0,1]$  for the  $d$ th dimension [1]. The  $P_i$  is the best position the  $i$ th particle has found so far and the  $P_n$  is the best previous position among a specified topology structure of the neighborhood. The  $w$  was first introduced by Shi and Eberhart [16], which decreases linearly from 0.9 to 0.4 during the evolution.

Generally, there are two major variants of PSO algorithms depending on the topology [17]. The *gbest* model (GPSO) shares information among the whole swarm and the *gbest* is the best position among all the particles. The *lbest* model (LPSO), whose neighborhood is constructed with a small group of particles. The ring topology proposed in [18] is a well-known local topology. In the ring topology, each particle connects with only two of the other particles in the ring and only makes use of the information from the two particles while updating its own position.

### B. CLPSO

The CLPSO was introduced by Liang et al. [13] to overcome the premature convergence by introducing a comprehensive learning (CL) strategy. The CL strategy enables the diversity of the swarm. In CLPSO, the particles choose different *pbest* values of different particles to update the velocity of different dimensions as:

$$v_{id} = \omega \times v_{id} + c \times r_d \times (p_{f(i)d} - x_{id}) \quad (3)$$

where  $\omega$  is the inertia weight as in (1),  $c$  is the accelerate coefficient fixed to be 1.49445, and  $r_d$  is a random value from 0 to 1. The  $f(i)$  is the particle index that used to guide the flying of the  $d$ th dimension, which can be any particle including the particle  $i$  itself.

In order to find the  $f(i)$  for each dimension, CLPSO firstly generates a random number  $r$ , then compare  $r$  with  $Pc_i$ , which is a probability to control learn from self or others. If  $r$  is smaller than  $Pc_i$ , then this dimension learns from others, otherwise learns from itself. When learning from others, CLPSO chooses two particles from the other particles and selects the one with better fitness as  $f(i)$ . If all the exemplars come from the particle  $i$  itself, randomly choose a dimension to learn from another particle. A particle will keep learning from its exemplars until it can not improve the solution quality for several generations which is called the refreshing gap  $m$ , then the new learning exemplars will be chosen again.

### C. APSO

In order to control the performance of PSO more objectively and effectively, Zhan et al. [8] introduced an evolutionary state estimation (ESE) technique to reflect the population and fitness diversity in their proposed adaptive PSO (APSO). In consideration of the fact that particles may disperse in different areas in the early stages and gather together to a locally or globally optimal area as the evolutionary process goes on, Zhan et al. [8] investigate the population distribution information in a PSO process and use ESE to the determine the evolutionary state for each generation.

With the evolutionary state information, the inertia weight

$\omega$  value is adaptively mapped to the search environment so as to balance the local and global search capabilities efficiently. Moreover, the acceleration coefficients  $c_1$  and  $c_2$  are adjusted according to the exploration state, exploitation state, convergence state, and jumping-out state adaptively.

Besides, in the APSO, an elitist learning strategy (ELS) is developed for the global leader  $gBest$  to improve itself when the algorithm is in convergence state. The ELS operation was designed to help the global leader  $gBest$  push itself out to a potentially promising region. The ELS randomly choose one dimension, then generate a new particle by mutating the  $gBest$ 's historical best position on the selected dimension. If the new particle is better than the  $gBest$ , then it will replace  $gBest$ , otherwise, it will replace the worst particle in the swarm.

#### D. OLPSO

In order to improve the learning strategy of PSO when searching in complex problem spaces, Zhan et al. [14] adjusted the conventional PSO by introducing a novel orthogonal experimental design (OED) mechanism in position learning in their proposed Orthogonal Learning Particle Swarm Optimization (OLPSO). In OLPSO, the traditional PSO learning mechanism is replaced by an Orthogonal learning (OL) strategy, which constructs an efficient and promising exemplar for a particle to learn from.

Without loss of generality, by using OED, each dimension is regarded as a factor. By this means, OLPSO combines information of the particle's historical best experience  $P_i$  and its neighborhood's best experience  $P_n$  to construct a guidance vector  $P_o=[p_{o1}, p_{o2}, \dots, p_{od}]$ . Then the  $i$ th particle's velocity can be updated by:

$$v_{id} = \omega \times v_{id} + c \times r_d \times (p_{od} - x_{id}) \quad (4)$$

where  $\omega$  is the inertia weight as in (1),  $c$  is the accelerate coefficient fixed to be 2.0, and  $r_d$  is a random value from 0 to 1. The  $p_{od}$  stands for  $p_{id}$  or  $p_{nd}$  according to the construct result of OED. That is,  $p_{od}$  just points to  $p_{id}$  or  $p_{nd}$  but is not the real exemplar.

In every generation, each particle  $i$  updates its own velocity and position with this promising and effective learning exemplar  $P_o$ . The  $P_o$  will be used as the exemplar for particle  $i$  until it cannot improve the solution quality for a certain number of generations which is called reconstruction gap  $G$ .

The OL operator can be implemented to PSO with any topological structures. As the results in [14] show that OLPSO based on local version PSO with ring topology can contribute to faster convergence speed and achieve promising solutions on both unimodal and multimodal problems, we adopt such OLPSO variant in this paper.

### III. COMPARISONS STUDY ON CEC 2005

The CEC 2005 test functions which are proposed as real-parameter optimization problems and include shifted and rotated test functions are selected in this study. The tested functions of CEC 2005 can be found at [19]. All the 25 functions are tested with these three PSOs in this study. The test functions can be divided into two groups, unimodal and multimodal functions. The definition of the first 12 functions

is based on some well-known classical benchmark functions (Sphere, Schwefel, Rosenbrock, Rastrigin, etc.). Among these 12 functions,  $F_1$  to  $F_5$  are unimodal while  $F_6$  to  $F_{12}$  are multi-modal. Moreover,  $F_{13}$  and  $F_{14}$  are expanded functions,  $F_{15}$  to  $F_{25}$  are the hybrid composition of several functions. For all the functions, the local optima is shifted to a non zero value to avoid exploitation of symmetry of the search space.

In this study, to make a fair comparison among the PSO algorithms, in all our experiments, dimension  $D$  is 30 and the number of total function evaluations (FEs) is assigned to be  $3 \times 10^5$ . According to the original references, the number of particles is set to 20 for APSO while is set to 40 for CLPSO and OLPSO. In APSO, the accelerations coefficients  $c_1$  and  $c_2$  are both initialized to 2.0 and adaptively controlled according to the evolutionary states. The accelerations coefficient  $c$  is set as 1.49445 for CLPSO and 2.0 for OLPSO, which is kept unchanged along the whole procedure. The inertia  $w$  is initialized to 0.9 at the beginning for the three PSOs, then linearly decreases to 0.4 at the end of the stage according to (3) for CLPSO and OLPSO while is adaptively adjusts for APSO. The refreshing gap  $m$  is set at 7 in CLPSO. The reconstruction gap  $G$  in OLPSO set to 5.

The APSO updates positions as (1) and (2). The CLPSO updates by (3). The OLPSO updates by (4). All the experiments are carried out in the same computer with Intel Pentium (R) Dual 2.40GHz CPU, 3.00GB memory, and a Win7 x32 operation system. To make the results more convincing, the algorithm is run for 30 independent times and the results of every run are recorded.

The results of CEC 2005 test functions for APSO, OLPSO, and CLPSO are shown in Table I. Error values ( $F(x)-F(x^*)$ ) are calculated as described in [19] and used to measure the performance. The best results on each function among these three algorithms are highlighted in **boldface** and the worst results are underlined. Fig. 1 plots the convergence progresses of the PSOs on some CEC 2005 test functions.

The simulations show that OLPSO performs best on most test functions. For the very simple unimodal function  $F_1$  (Shifted Sphere function) and the multimodal function  $F_9$  (Shifted Rastrigin's function with huge local optima number), OLPSO provides error values with the highest quality (0.00E+00). The experimental results also show that OLPSO outperforms the other two PSOs on all rotated hybrid composition functions ( $F_{16}$  to  $F_{25}$ ). This may benefit from the capacity of OL strategy in avoiding local optima and the ability of local version of PSO (because the OLPSO is based on the local version PSO) in preventing premature convergence to obtain the global optimum robustly.

The APSO shows significant advantages on unimodal functions except  $F_5$  (the Schwefel's Problem 2.6 with Global Optimum on Bounds).  $F_5$  is non-separable and scalable, if the initialization procedure initializes the population at the bounds, it will be solved easily [19]. In addition, APSO also performs best on some basic multimodal functions ( $F_6, F_7, F_8, F_{12}$ , and  $F_{13}$ ). However, the experimental results show that APSO almost performs worst on hybrid composition functions when comparing with OLPSO and CLPSO. In hybrid composition functions, different function's properties

are mixed together [19]. All of them have a huge number of local optima, and some of them are rotated or with noise. This makes them extremely complex functions. Hence, as Fig. 1(e) and Fig. 1(f) show, the faster convergence speed may affect the performance of APSO, resulting in early premature.

The CL strategy in CLPSO was designed to focus on maintaining the diversity of the swarm to obtain better

performance on multimodal problems. However, the disadvantage is the slow convergence on unimodal problems [13]. As Fig. 1 shows, the CLPSO converges most slowly among the three PSOs. Nevertheless, the ability of maintaining the population diversity helps it achieve similar performance on some complex hybrid composition functions as well as the OLPSO.

TABLE I. RESULTS OF APSO, OLPSO, AND CLPSO ON CEC 2005 BENCHMARK PROBLEMS

Functions	APSO			OLPSO			CLPSO		
	MEAN	STD	RANK	MEAN	STD	RANK	MEAN	STD	RANK
$F_1$	7.01E-14	2.45E-14	3	0.00E+00	0.00E+00	1	5.68E-14	0.00E+00	2
$F_2$	9.97E-13	1.79E-12	1	1.50E+01	1.23E+01	2	8.79E+02	1.79E+02	3
$F_3$	3.96E+05	1.59E+05	1	1.46E+07	5.33E+06	2	1.67E+07	4.66E+06	3
$F_4$	7.23E+01	6.02E+01	1	2.26E+03	9.70E+02	2	6.61E+03	1.14E+03	3
$F_5$	5.85E+03	1.45E+03	3	3.28E+03	5.54E+02	1	3.86E+03	5.32E+02	2
$F_6$	6.94E+00	1.68E+01	1	2.63E+01	2.50E+01	3	5.10E+00	5.43E+00	2
$F_7$	4.70E+03	2.34E-04	1	4.70E+03	1.61E-12	1	4.70E+03	6.78E-12	1
$F_8$	2.00E+01	2.97E-02	1	2.09E+01	6.90E-02	2	2.09E+01	5.46E-02	2
$F_9$	1.48E-13	5.90E-14	2	0.00E+00	0.00E+00	1	1.08E-11	1.02E-11	3
$F_{10}$	1.50E+02	6.25E+01	3	1.10E+02	3.12E+01	1	1.14E+02	1.50E+01	2
$F_{11}$	2.78E+01	3.16E+00	3	2.55E+01	2.95E+00	1	2.70E+01	1.71E+00	2
$F_{12}$	1.27E+04	1.70E+04	1	1.33E+04	6.95E+03	2	2.81E+04	6.59E+03	3
$F_{13}$	1.54E+00	4.05E-01	1	1.92E+00	3.28E-01	3	1.66E+00	5.68E-01	2
$F_{14}$	1.30E+01	5.24E-01	2	1.31E+01	2.57E-01	3	1.29E+01	1.72E-01	1
$F_{15}$	3.48E+02	1.50E+02	3	2.50E+02	9.21E+01	2	1.06E+02	5.34E+01	1
$F_{16}$	3.22E+02	1.41E+02	3	1.32E+02	3.74E+01	1	1.88E+02	3.11E+01	2
$F_{17}$	3.12E+02	1.32E+02	3	1.89E+02	3.25E+01	1	2.40E+02	3.89E+01	2
$F_{18}$	9.45E+02	2.19E+01	3	9.10E+02	1.82E+00	1	9.12E+02	1.35E+00	2
$F_{19}$	9.45E+02	5.63E+01	3	9.07E+02	2.03E+01	1	9.13E+02	1.12E+00	2
$F_{20}$	9.36E+02	4.18E+01	3	9.07E+02	2.03E+01	1	9.13E+02	1.28E+00	2
$F_{21}$	7.66E+02	3.23E+02	3	5.00E+02	2.86E-13	1	5.00E+02	4.14E-13	1
$F_{22}$	1.02E+03	5.65E+01	3	9.43E+02	1.35E+01	1	9.63E+02	1.36E+01	2
$F_{23}$	9.11E+02	3.04E+02	3	5.34E+02	3.59E-04	1	5.34E+02	1.39E-04	1
$F_{24}$	3.45E+02	3.76E+02	3	2.00E+02	2.89E-14	1	2.00E+02	6.19E-13	1
$F_{25}$	1.70E+03	1.74E+01	3	1.64E+03	5.59E+00	1	1.65E+03	3.77E+00	2
Ave. Rank	2.28			1.48			1.96		
Final Rank	3			1			2		

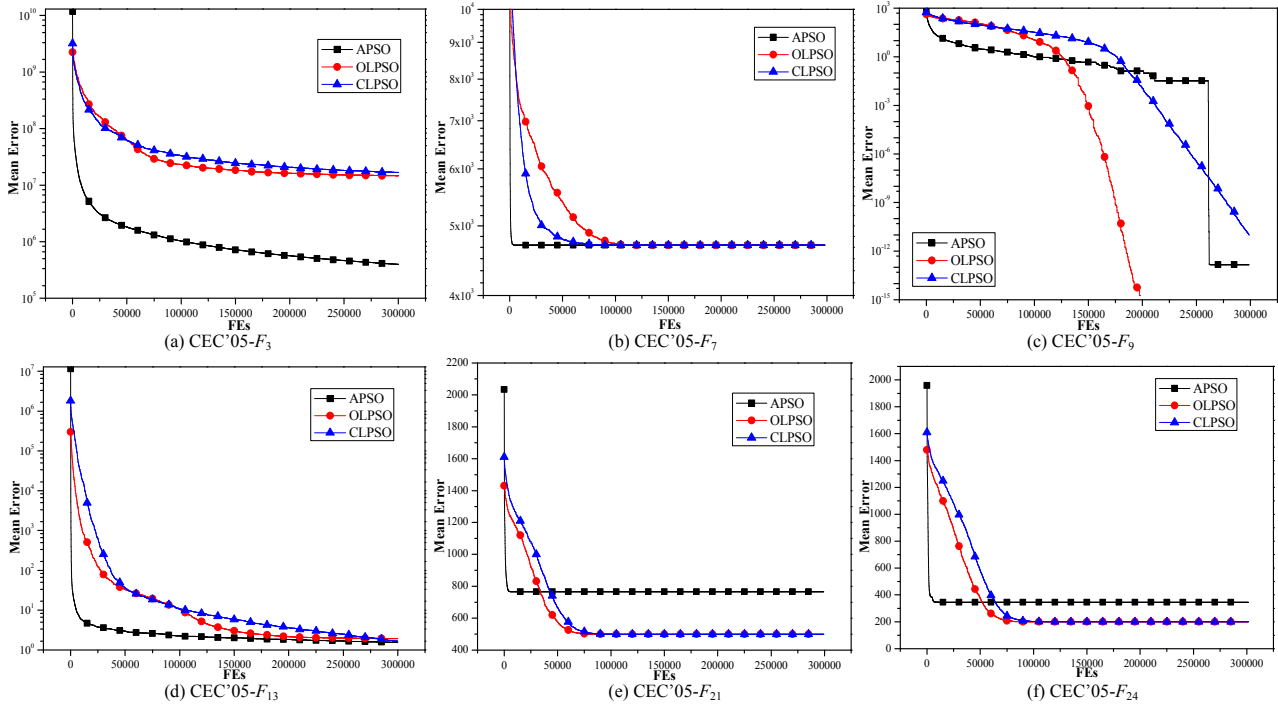


Fig. 1. Mean error curves for selected CEC 2005 benchmark problems

IV. COMPARISONS STUDY ON CEC 2014

Since the main focus of this paper is to compare the performance of APSO, OLPSO, and CLPSO in solving various complex functions, in order to examine whether one of them would outperform the others on a majority of the problems, we use a rather large number of benchmark problems from the CEC 2014. This test suit contains 30 benchmark functions that can be transformed into dynamic, niching composition, computationally expensive and many other classes. Therefore, they can evaluate the tested algorithms' characteristics on various problems. More details and analysis for the CEC 2014 Test Functions can be found at [20]. The 30 test functions can be divided into 4 groups, Unimodal Functions ( $F_1$  to  $F_3$ ), Simple Multimodal Functions ( $F_4$  to  $F_{16}$ ), Hybrid Function ( $F_{17}$  to  $F_{22}$ ) and Composition Functions ( $F_{23}$  to  $F_{30}$ ).

The results of CEC 2014 test functions for APSO, OLPSO, and CLPSO are listed in Tables II. Error values ( $F(x)-F(x^*)$ ) are calculated as described in [20] and used to measure the performance of the algorithms. The best results on each function among these three algorithms are highlighted in bold and the worse results are underlined. In addition, Fig. 2 graphically presents the comparison in terms of convergence characteristics of the evolutionary processes on several selected problems.

The simulations show that the performance of APSO is outstanding in comparison to the other two algorithms on all the three unimodal functions. APSO solves the Shifted and Rotated Rosenbrock's Function ( $F_4$ ) better than OLPSO and CLPSO.  $F_4$  has a very narrow valley from local optimum to global optimum, Fig. 2(b) shows that the APSO indeed jumps out of local optimal regions and achieves very good performance, mainly due to the help of ELS.  $F_{17}$  and  $F_{22}$  are hybrid functions constructed by some basic functions including  $F_1$ . Moreover,  $F_1$  holds the largest percentage among all the subcomponents for these two functions. Benefit from the good performance on  $F_1$  and also on other subcomponents, APSO shows advantages on  $F_{17}$  and  $F_{21}$  too. The composition function merges the properties of the sub-functions better and the local optimum which has the smallest bias value determines the global optimum [20]. Since the hybrid function  $F_{17}$  is also a basic function with the smallest bias for  $F_{29}$ , it decides more properties and influence than the other subcomponents and leads to the good performance of APSO on  $F_{29}$ .

As discussed in [14], the global version of OLPSO outperforms the local one on both accuracy and speed for the very simple unimodal functions but the local one is better for final solution accuracy for multimodal functions, this may explain why the local version of OLPSO used in this study does not show significant advantages on the three unimodal

TABLE II. RESULTS OF APSO, OLPSO, AND CLPSO ON CEC 2014 BENCHMARK PROBLEMS

Functions	APSO			OLPSO			CLPSO		
	MEAN	STD	RANK	MEAN	STD	RANK	MEAN	STD	RANK
$F_1$	<b>1.38E+05</b>	<b>9.59E+04</b>	1	6.12E+06	3.58E+06	2	<u>9.41E+06</u>	<u>3.02E+06</u>	3
$F_2$	<b>4.34E-03</b>	<b>1.02E-02</b>	1	<u>1.28E+03</u>	<u>1.48E+03</u>	3	2.76E+02	6.84E+02	2
$F_3$	<b>2.61E+02</b>	<b>4.10E+02</b>	1	3.23E+02	5.69E+02	2	<u>3.24E+02</u>	<u>2.79E+02</u>	3
$F_4$	<b>6.85E+00</b>	<b>2.03E+01</b>	1	<u>8.64E+01</u>	<u>2.22E+01</u>	3	8.07E+01	1.58E+01	2
$F_5$	<b>2.00E+01</b>	<b>1.88E-04</b>	1	2.03E+01	1.28E-01	2	<u>2.05E+01</u>	<u>4.95E-02</u>	3
$F_6$	<u>1.58E+01</u>	<u>3.53E+00</u>	3	<b>5.09E+00</b>	<b>1.48E+00</b>	1	1.43E+01	1.38E+00	2
$F_7$	<u>1.73E-02</u>	<u>2.08E-02</u>	3	<b>1.02E-13</b>	<b>3.47E-14</b>	1	6.68E-05	5.86E-05	2
$F_8$	8.86E-12	4.67E-11	2	<b>0.00E+00</b>	<b>0.00E+00</b>	1	<u>3.95E-11</u>	<u>5.48E-11</u>	3
$F_9$	<u>9.13E+01</u>	<u>2.46E+01</u>	3	<b>4.06E+01</b>	<b>7.02E+00</b>	1	6.11E+01	7.92E+00	2
$F_{10}$	7.92E-01	8.25E-01	2	<b>8.72E-02</b>	<b>2.04E-01</b>	1	<u>3.13E+00</u>	<u>1.52E+00</u>	3
$F_{11}$	2.74E+03	5.37E+02	2	<b>2.28E+03</b>	<b>4.66E+02</b>	1	<u>2.87E+03</u>	<u>2.73E+02</u>	3
$F_{12}$	<b>1.95E-01</b>	<b>7.17E-02</b>	1	2.28E-01	6.38E-02	2	<u>5.38E-01</u>	<u>7.21E-02</u>	3
$F_{13}$	<u>4.33E-01</u>	<u>9.22E-02</u>	3	<b>2.59E-01</b>	<b>3.20E-02</b>	1	3.32E-01	3.46E-02	2
$F_{14}$	<u>3.23E-01</u>	<u>1.10E-01</u>	3	<b>2.41E-01</b>	<b>2.66E-02</b>	1	2.78E-01	2.98E-02	2
$F_{15}$	<u>2.96E+01</u>	<u>4.03E+00</u>	3	<b>6.67E+00</b>	<b>1.62E+00</b>	1	8.62E+00	1.09E+00	2
$F_{16}$	<b>1.05E+01</b>	<b>8.21E-01</b>	1	<u>1.17E+01</u>	<u>5.48E-01</u>	3	1.06E+01	3.80E-01	2
$F_{17}$	<b>3.19E+04</b>	<b>2.14E+04</b>	1	7.98E+05	4.13E+05	2	<u>8.59E+05</u>	<u>3.58E+05</u>	3
$F_{18}$	<u>3.73E+03</u>	<u>5.23E+03</u>	3	3.58E+02	5.12E+02	2	<b>1.69E+02</b>	<b>5.85E+01</b>	1
$F_{19}$	<u>1.41E+01</u>	<u>1.84E+01</u>	3	<b>6.13E+00</b>	<b>8.20E-01</b>	1	8.35E+00	8.04E-01	2
$F_{20}$	<u>6.38E+03</u>	<u>4.86E+03</u>	3	5.58E+03	4.01E+03	2	<b>3.26E+03</b>	<b>1.70E+03</b>	1
$F_{21}$	<b>2.16E+04</b>	<b>1.31E+04</b>	1	<u>1.07E+05</u>	<u>8.33E+04</u>	3	8.08E+04	3.99E+04	2
$F_{22}$	<u>6.50E+02</u>	<u>2.42E+02</u>	3	2.20E+02	1.07E+02	2	<b>1.75E+02</b>	<b>7.77E+01</b>	1
$F_{23}$	<b>3.15E+02</b>	<b>1.15E-12</b>	1	<b>3.15E+02</b>	<b>1.23E-10</b>	1	<b>3.15E+02</b>	<b>4.86E-05</b>	1
$F_{24}$	<u>2.29E+02</u>	<u>4.73E+00</u>	3	<b>2.24E+02</b>	<b>5.47E-01</b>	1	2.25E+02	1.29E+00	2
$F_{25}$	<u>2.16E+02</u>	<u>5.81E+00</u>	3	2.09E+02	1.75E+00	2	<b>2.08E+02</b>	<b>1.19E+00</b>	1
$F_{26}$	<u>1.58E+02</u>	<u>6.05E+01</u>	3	<b>1.00E+02</b>	<b>4.44E-02</b>	1	<b>1.00E+02</b>	<b>7.35E-02</b>	1
$F_{27}$	<u>6.84E+02</u>	<u>2.11E+02</u>	3	<b>3.26E+02</b>	<b>3.80E+01</b>	1	4.17E+02	5.24E+00	2
$F_{28}$	<u>2.53E+03</u>	<u>8.16E+02</u>	3	<b>8.73E+02</b>	<b>2.97E+01</b>	1	8.98E+02	5.32E+01	2
$F_{29}$	<b>1.24E+03</b>	<b>5.02E+02</b>	1	<u>1.36E+03</u>	<u>2.82E+02</u>	3	1.29E+03	1.69E+02	2
$F_{30}$	<u>2.50E+03</u>	<u>6.63E+02</u>	2	<b>2.39E+03</b>	<b>5.99E+02</b>	1	<u>3.63E+03</u>	<u>1.00E+03</u>	3
AVE. Rank	2.13			<b>1.63</b>			2.1		
Final Rank	3			<b>1</b>			2		

functions. However, as the problem becomes more complex, the efficiency of the OL strategy can bring about improved performance of the OLPSO. For example, for the Shifted Rastrigin's Function ( $F_8$ ) with huge number of local optima, OLPSO has reliably found the minimum and reaches the 0.00E+00 error values. When solving problems that the local optima's number are huge and second better local optimum is far from the global optimum like  $F_{10}$  and  $F_{11}$ , by means of fully utilizing useful information of the learning exemplars, OLPSO yields the best performance among the three PSOs. Moreover, owing to the OL's ability of the constructing potential and efficient exemplars, the results also shows that though affected by the shift and rotation, OLPSO still performs better than the other two algorithms on most simple multimodal functions. This good performance of OLPSO is also observed on hybrid functions  $F_{18}$ ,  $F_{19}$ ,  $F_{22}$ , and most composition functions.

Although the results show that the CLPSO does not have significant advantages on the Unimodal Functions and the

Simple Multimodal Functions in comparing with APSO and OLPSO, CLPSO achieves the best or second best results on all the Hybrid Functions and the Composition Functions except  $F_{17}$ . As previously mentioned,  $F_{17}$  has certain concerned with  $F_1$ . As Fig. 2(a) and Fig. 2(e) show, owing to the slow convergence of CLPSO on unimodal problems like  $F_1$ , the CLPSO fails on  $F_{17}$ , so does OLPSO. The composition functions with complex asymmetrical landscapes are designed to merge the properties of the sub-functions better and maintain continuity around the global/local optima [20]. Solving these problems requires the algorithms to maintain the population diversity and avoid the prematurity. Since the CL strategy can effectively utilize all other particles' historical best information, it may be attractive to combine the CLPSO with some local search method to solve the real-world problems with fitness landscape we do not frequently know. Moreover, Fig. 2(d) shows that maintaining the population diversity is helpful for multimodal problems.

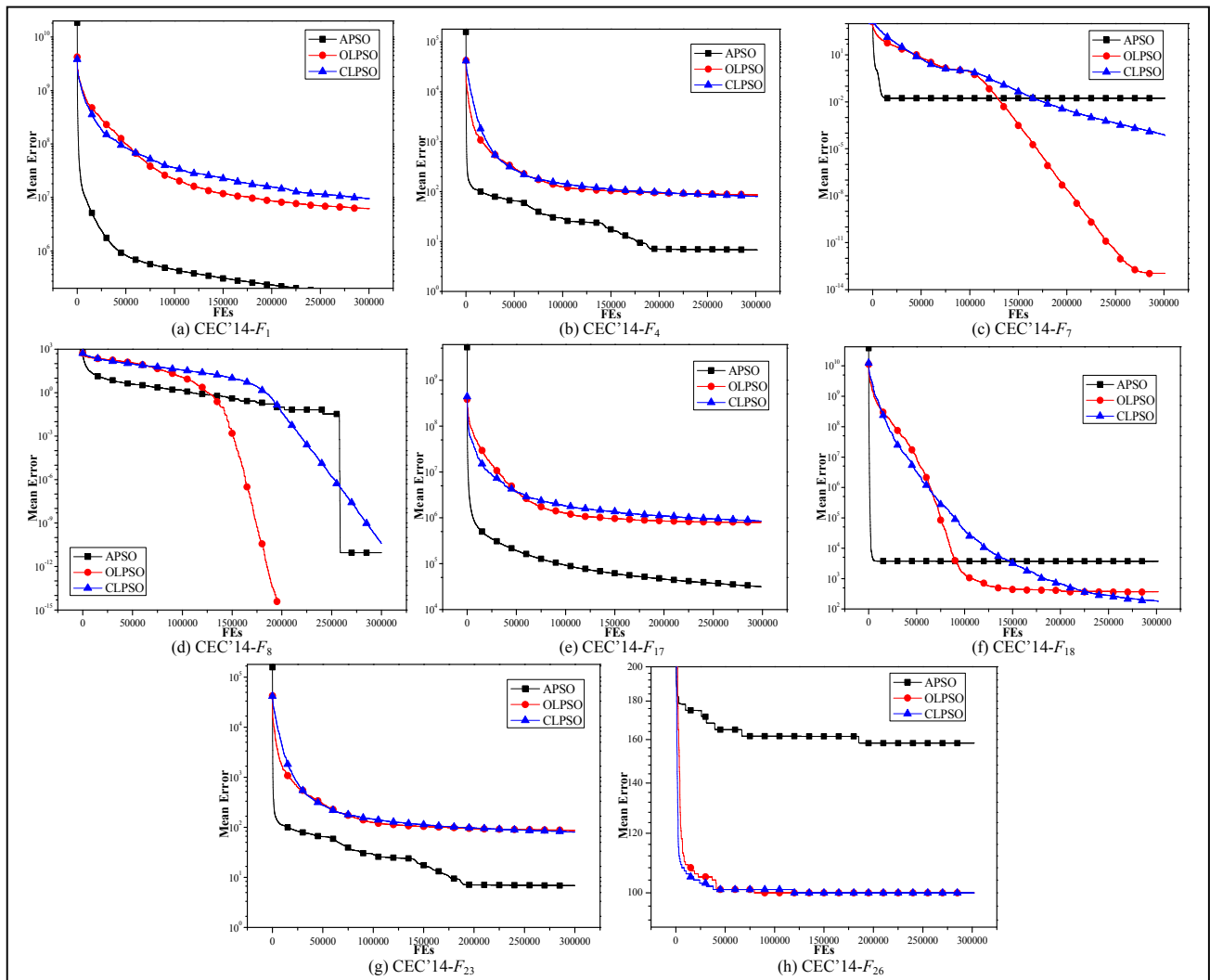


Fig. 2. Mean error curves for selected CEC 2014 benchmark problems

## V. CONCLUSION

In this paper, the performance of APSO, OLPSO, and CLPSO are compared on a large amount of complex benchmark problems. All the problems are shifted or rotated, and most of them are both shifted and rotated. Due to the simple concept and learning strategy of the traditional PSO, all these three improved PSOs are easy to understand and implement too. From the experiments and analysis, we can observe that the OLPSO yields the best performance in comparison to the other algorithms tested. Although the OLPSO may not be better than APSO at solving unimodal problems, the OL strategy can indeed make use of the information in swarm more effectively to generate better quality solutions frequently on complex multimodal problems. By effectively adjust the parameters, the novel APSO shows advantages on several benchmark problems, however, as the problems becomes more complex, the higher convergence speed APSO offers may degrade its performance and leads to the failure on composition problems. Moreover, CLPSO also shows potential performance on some extremely complex problems as well as OLPSO.

For details of the data, please contact the corresponding author Zhi-Hui Zhan ([zhazhh@mail.sysu.edu.cn](mailto:zhazhh@mail.sysu.edu.cn)) or visit the website: <http://www.ai.sysu.edu.cn/zhazhh>.

## REFERENCES

- [1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Networks*, 1995, pp. 1942–1948.
- [2] Y. H. Li, Z. H. Zhan, S. Lin, J. Zhang, and X. N. Luo, "Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems," *Information Sciences*, vol. 293, no. 1, pp. 370–382, 2015.
- [3] M. Shen, Z. H. Zhan, W. N. Chen, Y. J. Gong, J. Zhang, and Y. Li, "Bi-velocity discrete particle swarm optimization and its application to multicast routing problem in communication networks," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 12, pp. 7141–7151, Dec. 2014.
- [4] Z. H. Zhan, J. Li, J. Cao, J. Zhang, H. Chung, and Y. H. Shi, "Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 445–463, April. 2013.
- [5] Y. G. Fu, M. Y. Ding and C. P. Zhou, "Phase Angle-Encoded and Quantum-Behaved Particle Swarm Optimization Applied to Three-Dimensional Route Planning for UAVR," *IEEE Trans. Syst., Man, Cybern.A, System and Humans*, vol. 42, no.2, pp.511–526, Mar. 2012.
- [6] G.H. Zhao and Y. P. Du, "Research on airport apron service based on particle swarm optimization," *Management Science and Engineering (ICMSE), 2013 International Conference on. IEEE*, pp.484–488, Jul. 2013.
- [7] J. Zhang et al., "The application of improved particle swarm optimization algorithm in voltage stability constrained optimal power flow," *Measurement, Information and Control (ICMIC), 2013 International Conference on. IEEE*, vol 2, pp.1126–1130, Aug. 2013.
- [8] Z. H. Zhan, J. Zhang, Y. Li, and H. Chung, "Adaptive Particle swarm optimization," *IEEE Trans. Syst., Man, Cybern.B, Cybern.*, vol. 39, no. 6, pp. 1362–1381, Dec. 2009.
- [9] R. Mendes, J. Kennedy, and J. Neves, "The fully informed article swarm: simpler, maybe better," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 204–210, Jun. 2004.
- [10] B. Y. Qu, P. N. Suganthan, and S. Das, "A Distance-based Locally Informed Particle Swarm Model for Multi-modal Optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 387–402, 2013.
- [11] A. E. Muñoz, A. Hernández, and E. Villa, "Constrained optimization via particle evolutionary swarm optimization algorithm (PESO)," *In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2005*, pp. 209–216, 2005.
- [12] A. Ochoa, A. E. Muñoz-Zavala, and A. Hernández-Aguirre, "A Hybrid System Approach to Determine the Ranking of a Debutant Country in Eurovision," *Journal of Computers*, vol. 4, no. 8, pp. 713–720, 2009.
- [13] 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, Jun. 2006.
- [14] 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.
- [15] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
- [16] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE World Congr. Comput. Intell.*, 1998, pp. 69–73.
- [17] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *Proc. IEEE Swarm Intelligence Symposium*, 2007, pp.120–127.
- [18] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proc. IEEE Congr. Evol. Comput.*, Honolulu, HI, 2002, pp.1671–1676.
- [19] P. N. Suganthan, N. Hansen, J. J. Liang, et al, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," *KanGAL Report 2005005*, 2005.
- [20] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization," Nanyang Technological University, Singapore, Tech. Rep., 2014.