2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Partial Opposition-Based Adaptive Differential Evolution Algorithms: Evaluation on the CEC 2014 Benchmark Set for Real-parameter Optimization

Zhongyi Hu, Yukun Bao, and Tao Xiong
School of Management,
Huazhong University of Science and Technology,
Wuhan, P.R. China
Email: {huzhyi21,yukunbao,taoxiong}@hust.edu.cn

***Abstract*—Opposition-based Learning (OBL) has been reported with an increased performance in enhancing various optimization approaches. Instead of investigating the opposite point of a candidate in OBL, this study proposed a partial opposition-based learning (POBL) schema that focuses a set of partial opposite points (or partial opposite population) of an estimate. Furthermore, a POBL-based adaptive differential evolution algorithm (POBL-ADE) is proposed to improve the effectiveness of ADE. The proposed algorithm is evaluated on the CEC2014's test suite in the special session and competition for real parameter single objective optimization in IEEE CEC 2014. Simulation results over the benchmark functions demonstrate the effectiveness and improvement of the POBL-ADE compared with ADE.**

Keywords—opposition-based learning; differential evolution; real parameter; optimization.

I. INTRODUCTION

Opposition-based learning (OBL), originally introduced by Tizhoosh [1], tries to find a better candidate solution by simultaneously considering an estimate point and its corresponding opposite estimate. It has been proved that an opposite candidate solution can provide a higher chance of finding solutions that are closer to the global optimum one [2][3]. The concept of OBL has been applied to improve the performance meta-heuristic algorithms and machine learning algorithms [4], [5].

In [6], the convergence speed of evolutionary algorithm is accelerated by replacing the random initialization with the opposition-based population initialization. Further, [2] mathematically and experimentally proves this advantage when there is no prior knowledge about the solution. The benefits of the opposite of a candidate solution over random solutions is also shown intuitively based on an Euclidean distance-to-optimal solution proof in [7]. Recently, by considering the opposite individuals in the population initialization stage and generation jumping stage, OBL has been recently applied to accelerate various meta-heuristic algorithms, such as Differential Evolution (DE)[8]–[10], Particle Swarm Optimization (PSO) [11], [12], Biogeography-Based Optimization (BBO) [13]–[15], teaching and learning algorithm [16], gravitational search algorithm [17], Harmony Search (HS) [18], and Artificial Bee Colony (ABC)[19]. OBL are also employed to accelerate machine learning algorithms including reinforcement learning and backpropagation learning in neural networks, and Estimation of Distribution Algorithm (EDA). Opposition-based reinforcement learning (ORL) were proposed by considering opposite states and opposite actions[20]–[22]. The results demonstrate that ORL outperforms the reinforcement learning. Similarly, by considering the opposite transfer functions and opposite weights, the opposition-based neural networks were also proposed to improve their learning speed and accuracy [23]–[25].

Motivated by the idea of OBL, this study presents an improved OBL, namely partial opposition-based learning (POBL). Rather than only examining the opposite point of a candidate, the POBL is devised to compute partial opposite points (or partial opposite population) of an estimate. Further, a POBL-based adaptive DE algorithm is proposed to solve the numerical optimization problems in "CEC2014 Special Session and Competition on real parameter single objective optimization"[26]. In the proposed POBL-based adaptive DE, the Adaptive Differential Evolution (ADE) [27] that needs no parameters to be tuned is improved by POBL during the population initialization and generation jumping. Experimental simulations on benchmark functions show that POBL-ADE obtains better performance on the majority of the test problems compared with basic ADE and OBL-ADE.

The rest of the paper is structured as follows. Section II provides the overview of the ADE and the Opposition-based learning. Section III describes the proposed Partial opposition based learning and the POBL-based ADE. The experimental setting is depicted in Section IV. Section V reports the experimental results with discussions. Finally, Section V concludes this study with comments toward future research directions.

II. BACKGROUND

This section gives an overview of the Opposition-based Learning and Population's Variance-Based Adaptive Differential Evolution (ADE).

A. Opposition-Based Learning

Opposition-based learning (OBL) introduced by Tizhoosh [1] is a new concept in computational intelligence. When evaluating a solution to a given problem, simultaneously computing its opposite solution will provide another chance for finding a candidate solution which is closer to the global optimum.

The following shows the basic concepts of opposition-based learning, including opposite and quasi-opposite numbers and points.

Opposite number: given a real number x, with its range from a to b. The opposite number of x is defined by $: x^{\mathrm{o}} = a + b - x$.

By extending the definition of opposite to the higher dimensions, we can obtain the definition of opposite point:

Opposite point: given $X = (x_1, x_2, \cdots, x_D)$ is a point in a D dimensional space, where $x_i \in R, and\ x_i \in [a_i, b_i], i = 1,2,...,D$. Then, the opposite point $X^o = (x_1^o, x_2^o, \cdots, x_D^o)$ is defined as follows: $X_i^o = a_i + b_i - x_i$.

In [28], quasi-OBL is introduced and proved that the quasi-opposite point is more likely to closer than the opposite point to the solution. The Quasi-opposite number x^{qo} is defined as the random number between the opposite number x^{o} and the center c of the search space. Mathematically, x^{qo} is expressed as:

$$x^{\mathrm{qo}} = rand\left((a+b)/2, x^{\mathrm{o}}\right) \tag{1}$$

And the quasi-opposite point in a d-dimensional space is given by:

$$X_i^{\mathrm{qo}} = rand\left((a_i+b_i)/2, x_i^o\right),\ i = 1,2,...,D \tag{2}$$

In the opposition-based learning (OBL), both the point and its opposite point (or quasi-opposite point) are evaluated simultaneously, and the better one is selected to continue the evolution.

B. Population's Variance-Based Adaptive Differential Evolution (PVADE)

Differential Evolution (DE), first proposed by Storn and Price [29], is a branch of evolutionary algorithm (EA). Different from other EAs, DE perturb the mutation by considering the difference of two individuals. It has shown better performance in terms of convergence speed, computation complexity, and robustness, than many other meta-heuristic methods such as genetic algorithm (GA) particle swarm optimization (PSO) [30]. Unfortunately, the success of DE is crucially depends on appropriately selecting the control parameters such as population size NP, crossover rate (CR) , and scaling factor F[31], [32].

Instead of performing a time-consuming trial-and-error tuning for these control parameters involved in DE, [27] proposed an adaptive differential evolution (ADE). In the ADE, the related parameters including scaling factor, crossover rate and quasi-oppositional probability are adaptively changed based on population's variance information. Besides, different scaling factors for each dimension is employed in the ADE instead of only one scaling factors as in classical DE. The performance of the ADE is validated on the test suite for real parameter single objective optimization which is specially designed in IEEE CEC2013. Considering its adaptive mechanism and performance reported in CEC2013, it is employed in this study.

We refer to [27] for a detailed statement of the main concepts of population variance based adaptive DE.

III. PARTIAL OPPOSITION-BASED LEARNING AND POBL-BASED ADE

This section proposes the partial opposition-based learning (POBL) and the POBL based ADE.

A. Partial Opposition-based Learning (POBL)

For OBL in multi-dimensional space, only the complete opposite point $X^o = (\mathrm{x}_1^o, \mathrm{x}_2^o, \cdots, \mathrm{x}_D^o)$, in which each dimension is opposite to the original value is considered. Inspired by the opposition concept and to enhance the exploration of the algorithm, the partial opposite point and partial opposition-based learning is proposed.

In a multi-dimensional space, the partial opposite point is $x^{po} = (x_1, x_2^o, \cdots, x_D^o)$,where only some dimensions are opposite to the original dimension. So, we can define the partial opposite population $\{X^{po}\}$ as follows:

$$X^{po} = \left\{X_1^{po}, X_2^{po}, ..., X_D^{po}\right\} = \begin{Bmatrix} (x_1 & x_2^o & x_3^o & \cdots & x_D^o) \\ (x_1^o & x_2 & x_3^o & \cdots & x_D^o) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (x_2^o & x_2^o & x_3^o & \cdots & x_D) \end{Bmatrix} \tag{3}$$

where, x_i^o is the opposition value of x_i . In (3), for each partial opposite point X_i^{po} , only one dimension stay the same with original value, so it can be named as the 1st-degree partial opposition. Based on the partial opposite population, we proposed the partial opposition-based learning (POBL) as follows.

Given $X = (x_1, x_2, \cdots, x_D)$ is a point in D dimensional space, and $f(.)$ is the fitness function used to evaluate the candidate solution. The opposite point $X^o = (x_1^o, x_2^o, \cdots, x_D^o)$ and some randomly selected partial opposite points are examined. The best point is selected to compare with the original point. Only if its fitness value is better than the original point *X*, it is applied to replace with original point. Otherwise, we continue with *X*. In the proposed POBL, both the original point, the opposite point and some partial opposite points are evaluated simultaneously in order to get the better solution.

B. POBL-based ADE

Based on proposed POBL and adaptive DE, this subsection states the POBL-based ADE as follows.

Step 1. Population initialization: randomly initialize all the individuals within the range of lower and upper bounds of the problem.

Step 2. Calculate the adaptive scaling factors: $Fm_{,j}$ is defined as follows in (4):

$$Fm_{,j} = \begin{cases} Fscaling(j), & \text{rand}(0,1) > 0.2 \\ 0.7*(t/t_{\max}) + 0.2, & otherwise \end{cases} \tag{4}$$

where,

$Fscaling(j) = \max\{\min(0.6, 1 - pop\_diversity), 0.3\}$

$pop\_diversity = (\text{D}/20) * pop\_var(t) / max\_var$

$pop\_var(t) = var(\text{x}_{\text{i}}(\text{t}))$

$max\_var = \max(pop\_var)$

As shown in (4), the scaling factors in ADE is different in each dimension. And their values are generated by using the population diversity for each dimension based on population variance.

Step 3. Mutation operation: based on three randomly selected individuals, the mutate vector is generated with adaptive scaling factors according to the following equation:

$$z_{i,j}(t+1) = x_{r1,j}(t) + Fm_j * \left(x_{r2j}(t) - x_{r3,j}(t)\right) \tag{5}$$

Step 4. Calculate the adaptive crossover rate (*CR*) and opposition probability: crossover rate and the probability of quasi-opposition (*prob_qop*) in ADE are also tuned adaptively based on the population variance, by (6):

$$CR = \begin{cases} 0.1 * \max(1 - pop\_diversity), & WI > 90 \\ \max(1 - pop\_diversity), & \text{otherwise} \end{cases}$$
$$prob\_qop = \begin{cases} (D/60) * pop\_diversity, & WI > 90 \\ (D/33.33) * pop\_diversity, & \text{otherwise} \end{cases} \tag{6}$$

where *WI* counts the number of generations without any improvement of the best solution in ADE.

Step 5. Crossover operation: based on the mutant vector and adaptive *CR*, a trial individual is generated.

Step 6. Partial opposition-based Learning: the POBL is executed with the probability *prob_qop* for the each solutions. In the each solution's POBL, each partial opposite point is selected with a probability *prob_POBL*. The *prob_POBL* is experimentally set as a random value between 0.1 and 0.3.

Step 7. repeat the loop from Step 2 until the termination condition is satisfied.

IV. EXPERIMENTAL SETUP

The performance of the proposed algorithm was examined using CEC'14 test suite in the special session and competition on the single objective real-parameter numerical optimization(http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2014/CEC2014.htm). The CEC'14 test suite consists of 30 benchmark functions, which comprise three unimodal functions (f1-f3), thirteen simple multimodal functions (f4-f16), six hybrid functions (f17-f22), and eight composition functions (f23-f30). These benchmark problems are specially designed with several novel features such as novel basic problems, composing test problems by extracting features dimension-wise from several problems, graded level of linkages, rotated trap problems, etc..

As asked by the organizer, experiments were carried out for all these functions with dimension (D) equals to 10, 30, 50, and 100. For each function, the maximum number of function evaluations (MaxFES) is set to 10000*D. Search space for each function was $[-100,100]^D$. Each experiment terminates when the MaxFES is reached or the error value is smaller than 1E-08. Detail descriptions of the CEC14' test suite and evaluation criteria in this special session is provided in [26].

In the proposed POBL-based ADE, parameters including scaling factors (*F*) and crossover rate (*CR*) in ADE is adaptively tuned based on population variance. The probability of quasi-opposition (*prob_qop*) is also adaptively tuned. The population size is set with the same to [27], that is, population size equals to 50, 100, 150, and 200 for each function with the dimension of 10, 30, 50, and 100, respectively. The only free parameter in POBL-ADE is the selective probability (prob_POBL) of each partial opposite point. Extensive experiments are conducted and finally, a random value between 0.1 and 0.3 is applied for *prob_POBL*.

All experiments were executed in Matlab 2012a on an Intel® Core 2 Duo CPU, 2.33GHZ, with the 3GB RAM, 32-bit Windows 7 system.

V. RESULTS AND DISCUSSIONS

The complexity of the proposed algorithm is presented in Table I. It is easy to conclude that more computational cost is required with the increasing number of dimensions for the test suite.

Median convergence characteristic of the proposed POBL_ADE on some selected functions for dimensionality 100 are shown in Fig.1 to 6. The y-axis gives the median functions error value. The x-axis shows 14 specified functions evaluation checkpoint at (0.01, 0.02, 0.03, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.8, 1.0)*MaxFES. The figures show that the proposed approach has been convergent in most functions. The details convergence data for other dimension functions are given in the technical report.

TABLE I. COMPUTATIONAL COMPLEXITY OF THE PROPOSED ALGORITHM

Dimension	$T0$	$T1$	$\hat{T}2$	$(\hat{T}2-T1)/T0$
D=10	0.1917	2.4049	2.4674	0.3260
D=30		3.6987	3.9641	1.3845
D=50		5. 1978	5.4766	1.4862
D=100		10.9471	11.2991	1.8362

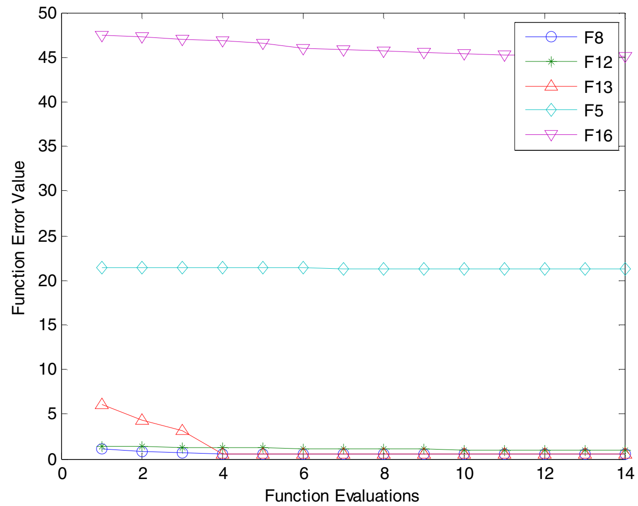

Fig. 1. Median convergence characteristic for 100D

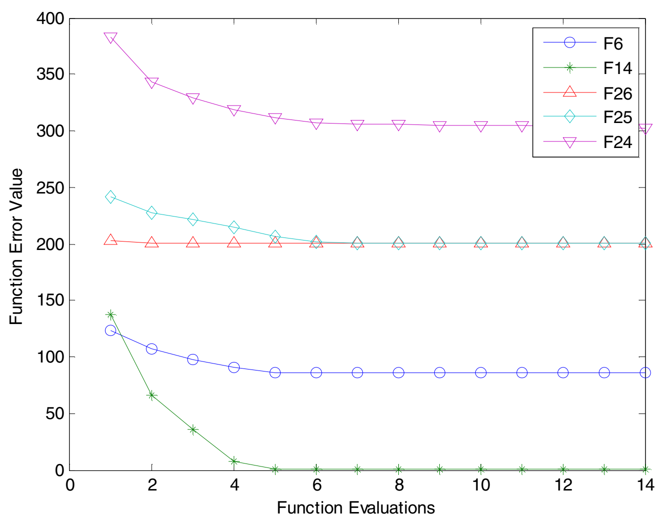

Fig. 2. Median convergence characteristic for 100D

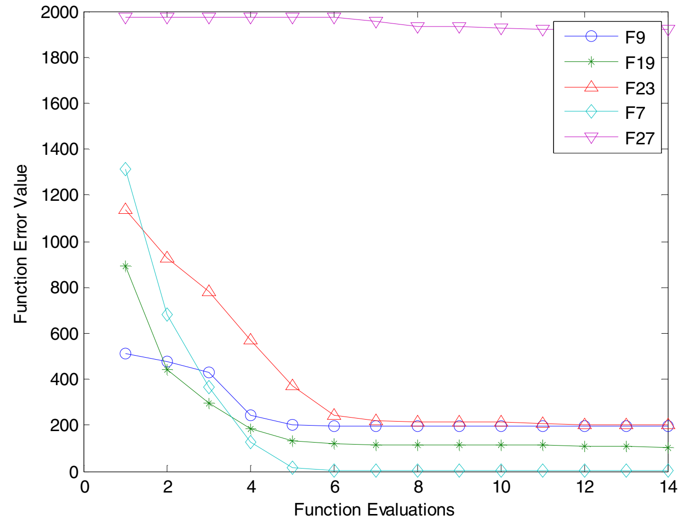

Fig. 3. Median convergence characteristic for 100D

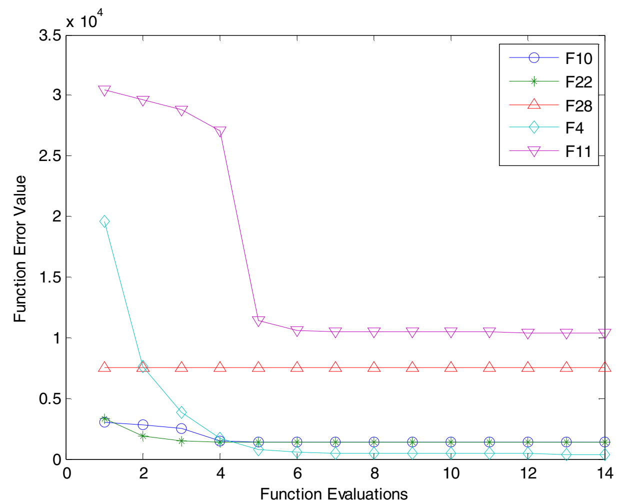

Fig. 4. Median convergence characteristic for 100D

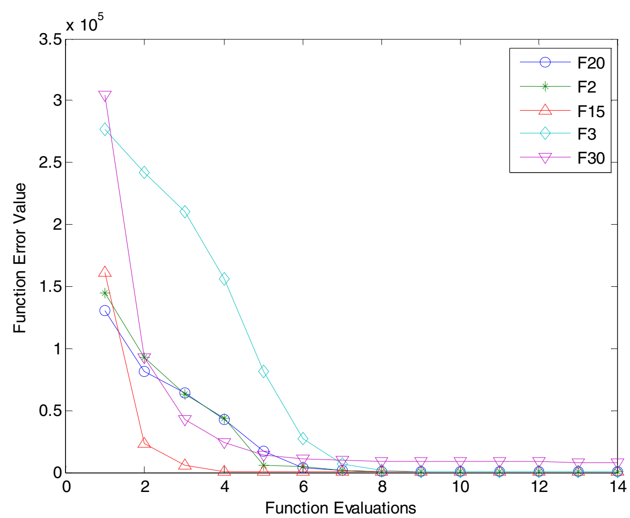

Fig. 5. Median convergence characteristic for 100D

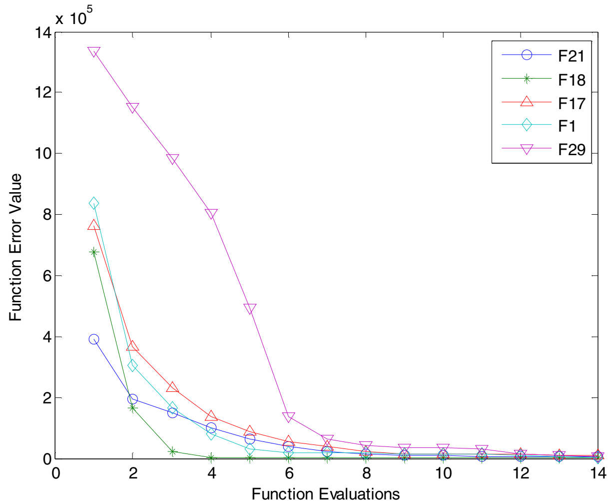

Fig. 6. Median convergence characteristic for 100D

Table II, III, IV,V show the error values obtained by POBL-ADE in 51 independent runs for each function for dimension 10, 30, 50, and 100. For the purpose of the competition, the error values in the table including the best, worst, median, mean, and the standard deviation values. As the organizers noticed, the error values smaller than 1.00E-08 can be set as zero, we labeled these values in Bold. While the values that is a little large (larger than 1.00E+03) is labeled with underline.

As shown in Table II, the POBL-ADE succeeded in finding the true optimum value at least one time for only unimodal functions f1. While for unimodal functions f1,f2, and simple multimodal function f6, the best error value of the algorithm is very close to zero, they are all smaller than 1.00E-03. At the same time, the algorithm has the ability in obtaining the reasonable error value (i.e., best error is smaller than 1.00E+01) for almost all the unimodal functions, simple multimodal functions, and hybrid functions, except f10 and f17. While for all the composition functions, the best error value is still larger than 1.00E+02. From the comparisons in terms of best error value, we can conclude that the composition functions are more difficult to solve. However, it is worth noting that, for unimodal functions f1 and f2, they are, to some extent, more instable compared with simple multimodal functions, hybrid functions, and even composition functions. While unimodal functions f1, f2, and composition functions f29 have the biggest values (larger than 1.00E+03) than others in terms of std. values, the rest functions such as f10, f11, all hybrid functions (except f19), and composition functions (except f23 and f26) are larger than 1.00E+01. Overall, the proposed algorithm has obtained both good mean error value (smaller than 1.00E+00) and std. value (smaller than 1.00E-01) for unimodal functions f3, simple multimodal functions f7, and f12-f15.

Similarly, Table III presents summarized results for 30-dimention problem. The results are, to some extent, consistent with those obtained in 10-dimension cases. To be noticed, for 30-dimention problem in Table III, POBL-ADE can find the true optimum value for unimodal functions f2, f3; and simple multimodal functions f7, at least one time. Meanwhile, for the unimodal function f3, the median error value is equal to 0, and the worst, mean, and std error values are all smaller than 1E-08, which indicates that POBL-ADE has been successful in solving the f3 in all 51 runs. As for the 50D and 100D problems in Table IV and Table V, the mean errors for almost all functions are larger than1.00E+00.

TABLE II. RESULTS FOR 10D

f.	Best	Worst	Median	Mean	Std.
1	3.34E-03	1.72E+05	5.55E+02	1.62E+04	3.75E+04
2	1.84E-05	1.14E+04	5.45E+02	2.27E+03	3.27E+03
3	**0.00E+00**	1.04E-02	4.84E-07	5.74E-04	1.91E-03
4	2.83E-01	4.18E+01	3.48E+01	2.55E+01	1.41E+01
5	3.57E+00	2.01E+01	2.01E+01	1.91E+01	3.60E+00
6	6.48E-04	3.16E+00	8.95E-01	1.04E+00	7.83E-01
7	1.72E-02	1.20E+00	1.06E-01	1.63E-01	1.94E-01
8	2.98E+00	1.69E+01	5.99E+00	7.81E+00	3.87E+00
9	9.95E-01	1.69E+01	6.96E+00	7.63E+00	4.07E+00
10	2.19E+01	5.65E+02	1.59E+02	1.53E+02	1.16E+02
11	3.60E+00	5.83E+02	1.58E+02	2.08E+02	1.43E+02
12	1.27E-01	3.65E-01	2.80E-01	2.69E-01	5.85E-02
13	5.73E-02	2.34E-01	1.21E-01	1.31E-01	4.69E-02
14	5.49E-02	5.38E-01	2.35E-01	2.60E-01	1.27E-01
15	1.71E-01	1.41E+00	6.87E-01	7.12E-01	2.41E-01
16	4.00E-01	2.64E+00	1.43E+00	1.41E+00	5.21E-01
17	2.03E+01	6.06E+02	2.17E+02	2.57E+02	1.63E+02
18	1.34E+00	1.33E+02	2.11E+01	3.32E+01	3.36E+01
19	1.93E-01	5.78E+00	1.81E+00	2.09E+00	1.09E+00
20	1.44E+00	5.40E+01	7.93E+00	1.26E+01	1.18E+01
21	2.21E-01	4.25E+02	7.55E+01	1.03E+02	1.13E+02
22	4.95E-01	1.63E+02	2.07E+01	3.00E+01	3.38E+01
23	3.29E+02	3.29E+02	3.29E+02	3.29E+02	2.62E-04
24	1.06E+02	2.07E+02	1.17E+02	1.24E+02	2.41E+01
25	1.21E+02	2.02E+02	2.01E+02	1.86E+02	2.66E+01
26	1.00E+02	1.00E+02	1.00E+02	1.00E+02	4.96E-02
27	1.60E+00	4.01E+02	3.25E+02	2.56E+02	1.68E+02
28	3.57E+02	6.01E+02	4.04E+02	4.23E+02	5.58E+01
29	2.22E+02	3.88E+06	2.42E+02	3.55E+05	9.40E+05
30	4.74E+02	1.33E+03	5.99E+02	6.38E+02	1.64E+02

TABLE III. RESULTS FOR 30D

f.	Best	Worst	Median	Mean	Std.
1	1.50E+03	6.38E+04	1.26E+04	1.60E+04	1.22E+04
2	**0.00E+00**	3.15E+03	6.91E-01	3.14E+02	7.52E+02
3	**0.00E+00**	**3.28E-08**	**0.00E+00**	**6.43E-10**	**4.59E-09**
4	3.16E-03	9.95E+01	7.08E+01	6.34E+01	2.63E+01
5	2.05E+01	2.07E+01	2.06E+01	2.06E+01	5.11E-02
6	1.88E+00	8.50E+00	5.01E+00	5.19E+00	1.64E+00
7	**0.00E+00**	8.10E-02	2.21E-02	2.37E-02	2.31E-02
8	3.75E+01	9.30E+01	5.39E+01	5.59E+01	1.10E+01
9	6.35E+01	1.06E+02	8.43E+01	8.46E+01	9.06E+00
10	8.94E+02	3.35E+03	2.11E+03	2.17E+03	4.92E+02
11	2.92E+03	4.56E+03	3.91E+03	3.86E+03	3.52E+02
12	6.11E-01	1.25E+00	9.58E-01	9.51E-01	1.35E-01
13	1.56E-01	4.16E-01	2.79E-01	2.86E-01	6.10E-02
14	1.06E-01	3.08E-01	2.28E-01	2.26E-01	4.28E-02
15	4.73E+00	9.45E+00	7.81E+00	7.73E+00	1.04E+00
16	9.09E+00	1.11E+01	1.06E+01	1.04E+01	4.58E-01
17	9.42E+01	2.41E+03	1.08E+03	1.10E+03	4.14E+02
18	5.89E+01	2.12E+02	9.83E+01	1.10E+02	3.81E+01
19	3.57E+00	6.88E+01	6.56E+00	8.88E+00	1.21E+01
20	8.94E+00	1.06E+02	3.81E+01	3.89E+01	2.21E+01
21	1.25E+02	9.00E+02	3.22E+02	3.86E+02	1.91E+02
22	1.41E+02	4.11E+02	2.23E+02	2.31E+02	8.16E+01
23	3.15E+02	3.15E+02	3.15E+02	3.15E+02	1.16E-07
24	2.00E+02	2.36E+02	2.23E+02	2.22E+02	7.48E+00
25	2.00E+02	2.09E+02	2.03E+02	2.04E+02	3.22E+00
26	1.00E+02	2.00E+02	1.01E+02	1.39E+02	4.91E+01
27	3.26E+02	5.34E+02	4.03E+02	4.21E+02	4.64E+01
28	7.43E+02	1.92E+03	8.96E+02	9.16E+02	1.63E+02
29	4.70E+02	1.72E+07	7.71E+02	3.39E+05	2.41E+06
30	5.00E+02	3.35E+03	1.23E+03	1.29E+03	5.14E+02

TABLE IV. RESULTS FOR 50D

f.	Best	Worst	Median	Mean	Std.
1	1.45E+03	1.46E+04	6.08E+03	6.54E+03	2.95E+03
2	4.13E-03	5.63E+01	8.91E+00	1.37E+01	1.41E+01
3	3.68E-02	2.43E+00	3.16E-01	5.22E-01	5.40E-01
4	1.32E-01	1.55E+02	1.07E+02	1.03E+02	3.97E+01
5	2.09E+01	2.11E+01	2.10E+01	2.10E+01	4.13E-02
6	1.25E+01	2.83E+01	1.88E+01	1.93E+01	3.62E+00
7	3.39E-06	1.96E-01	1.48E-02	2.88E-02	4.11E-02
8	5.57E-03	2.15E-02	1.55E-02	1.46E-02	4.15E-03
9	6.47E+01	2.68E+02	2.16E+02	2.16E+02	2.90E+01
10	2.36E+02	7.78E+02	5.12E+02	4.97E+02	1.59E+02
11	5.27E+03	8.58E+03	7.20E+03	7.20E+03	6.09E+02
12	1.43E+00	2.41E+00	1.90E+00	1.85E+00	2.00E-01
13	3.19E-01	5.59E-01	4.17E-01	4.21E-01	5.68E-02
14	2.12E-01	3.53E-01	2.82E-01	2.83E-01	3.37E-02
15	1.64E+01	3.46E+01	2.29E+01	2.30E+01	3.07E+00
16	1.89E+01	2.09E+01	2.00E+01	2.00E+01	4.11E-01
17	1.45E+03	6.99E+03	2.82E+03	3.08E+03	1.22E+03
18	7.53E+01	3.55E+03	1.59E+02	4.14E+02	8.36E+02
19	1.09E+01	2.32E+01	1.67E+01	1.65E+01	3.20E+00
20	1.06E+02	2.80E+02	1.79E+02	1.85E+02	4.20E+01
21	4.76E+02	1.93E+03	1.45E+03	1.42E+03	2.86E+02
22	2.64E+02	8.05E+02	5.18E+02	4.83E+02	1.38E+02
23	3.44E+02	3.44E+02	3.44E+02	3.44E+02	1.16E-03
24	2.47E+02	2.79E+02	2.57E+02	2.59E+02	6.47E+00
25	2.00E+02	2.00E+02	2.00E+02	2.00E+02	1.40E-05
26	1.01E+02	2.00E+02	2.00E+02	1.86E+02	3.45E+01
27	6.52E+02	1.03E+03	8.56E+02	8.52E+02	9.67E+01
28	1.17E+03	3.88E+03	2.11E+03	2.25E+03	6.53E+02
29	2.07E+01	6.57E+06	2.79E+01	2.56E+05	1.28E+06
30	9.94E+03	2.60E+04	1.12E+04	1.17E+04	2.31E+03

TABLE V. RESULTS FOR 100D

f.	Best	Worst	Median	Mean	Std.
1	1.43E+03	4.87E+03	2.91E+03	3.01E+03	8.10E+02
2	4.61E-01	6.48E+01	1.34E+01	1.54E+01	1.10E+01
3	3.44E+00	5.97E+01	1.54E+01	1.79E+01	1.17E+01
4	2.34E+02	4.10E+02	3.20E+02	3.24E+02	4.62E+01
5	2.12E+01	2.13E+01	2.13E+01	2.13E+01	2.49E-02
6	7.42E+01	1.02E+02	8.60E+01	8.68E+01	6.57E+00
7	6.04E-04	1.82E-01	5.46E-03	1.93E-02	3.06E-02
8	3.74E-01	5.89E-01	5.02E-01	5.05E-01	4.20E-02
9	1.43E+02	2.63E+02	1.97E+02	1.96E+02	2.59E+01
10	1.20E+03	1.56E+03	1.36E+03	1.36E+03	7.47E+01
11	8.11E+03	1.27E+04	1.04E+04	1.03E+04	1.02E+03
12	7.80E-01	1.07E+00	9.59E-01	9.47E-01	6.68E-02
13	4.19E-01	6.15E-01	5.22E-01	5.16E-01	5.01E-02
14	1.09E-01	1.46E-01	1.27E-01	1.28E-01	9.01E-03
15	2.18E+01	3.69E+01	2.94E+01	2.94E+01	2.98E+00
16	4.35E+01	4.65E+01	4.51E+01	4.52E+01	6.02E-01
17	3.49E+03	2.07E+04	9.61E+03	1.02E+04	3.69E+03
18	1.89E+02	3.51E+03	5.62E+02	7.91E+02	6.56E+02
19	3.87E+01	1.75E+02	1.02E+02	9.37E+01	2.72E+01
20	2.15E+02	5.49E+02	3.17E+02	3.29E+02	6.89E+01
21	1.83E+03	1.74E+04	4.63E+03	5.07E+03	2.63E+03
22	7.82E+02	2.33E+03	1.35E+03	1.40E+03	3.76E+02
23	2.00E+02	3.48E+02	2.00E+02	2.23E+02	5.44E+01
24	2.01E+02	3.29E+02	3.03E+02	2.67E+02	5.43E+01
25	2.00E+02	2.00E+02	2.00E+02	2.00E+02	6.69E-04
26	2.00E+02	2.00E+02	2.00E+02	2.00E+02	1.69E-02
27	1.54E+03	2.39E+03	1.92E+03	1.96E+03	2.01E+02
28	4.81E+03	1.01E+04	7.48E+03	7.49E+03	1.30E+03
29	2.71E+03	2.36E+05	6.64E+03	1.25E+04	3.22E+04
30	4.97E+03	1.29E+04	7.58E+03	7.83E+03	1.65E+03

Furthermore, to verify the performance of the proposed POBL in improving the ADE, basic ADE is also conducted in the test suite with the dimension of 50 and 100. The mean values of each algorithm are shown in Table VI. Better performance for each function is labeled in bold. It is shown that POBL-ADE gains better results than ADE in almost 2/3 functions. It confirms the improvement obtained by the use of proposed POBL.

TABLE VI. COMPARISON RESULTS BETWEEN POBL-ADE AND ADE IN 50D AND 100D

f.	50D		100D	
	POBL-ADE	ADE	POBL-ADE	ADE
1	**6.54E+03**	7.95E+03	**3.01E+03**	3.87E+03
2	**1.37E+01**	1.93E+01	**1.54E+01**	1.83E+01
3	5.22E-01	**5.20E-01**	**1.79E+01**	2.33E+01
4	1.03E+02	**9.80E+01**	**3.24E+02**	3.79E+02
5	**2.10E+01**	2.50E+01	**2.13E+01**	2.71E+01
6	1.93E+01	**1.47E+01**	**8.68E+01**	1.08E+02
7	2.88E-02	**2.47E-02**	1.93E-02	**1.90E-02**
8	**1.46E-02**	2.24E-02	5.05E-01	**3.68E-01**
9	2.16E+02	**1.82E+02**	**1.96E+02**	2.37E+02
10	**4.97E+02**	6.61E+02	**1.36E+03**	1.45E+03
11	**7.20E+03**	1.03E+04	**1.03E+04**	1.09E+04
12	1.85E+00	**1.78E+00**	9.47E-01	**9.15E-01**
13	4.21E-01	**4.14E-01**	**5.16E-01**	7.22E-01
14	2.83E-01	**2.54E-01**	1.28E-01	**1.15E-01**
15	**2.30E+01**	2.60E+01	**2.94E+01**	3.50E+01
16	**2.00E+01**	2.52E+01	4.52E+01	**3.99E+01**
17	3.08E+03	**2.22E+03**	**1.02E+04**	1.19E+04
18	4.14E+02	**3.43E+02**	**7.91E+02**	8.36E+02
19	**1.65E+01**	2.14E+01	**9.37E+01**	1.40E+02
20	1.85E+02	**1.81E+02**	3.29E+02	**2.52E+02**
21	**1.42E+03**	1.60E+03	**5.07E+03**	5.61E+03
22	**4.83E+02**	6.70E+02	1.40E+03	**1.00E+03**
23	**3.44E+02**	4.39E+02	2.23E+02	**1.90E+02**
24	**2.59E+02**	3.90E+02	2.67E+02	**1.95E+02**
25	**2.00E+02**	3.00E+02	**2.00E+02**	2.14E+02
26	**1.86E+02**	2.33E+02	**2.00E+02**	2.32E+02
27	**8.52E+02**	1.14E+03	**1.96E+03**	2.58E+03
28	**2.25E+03**	3.72E+03	7.49E+03	**6.39E+03**
29	**2.56E+05**	2.85E+05	**1.25E+04**	1.41E+04
30	1.17E+04	**8.97E+03**	7.83E+03	**6.96E+03**

VI. CONCLUSION

The opposition-based learning has been recently applied to enhance many evolutionary algorithms and machine learning algorithms. Inspired by OBL, this study proposed a partial opposition-based learning that simultaneously considering the original point, the opposite point and some partial opposite points. POBL was then applied to an adaptive differential evolution algorithms. According to the reported results in 10D, 30D, 50D, and 100D problems in CEC14's suite, the POBL was capable to enhance the performance of ADE compared with basic ADE. Besides, this study is just a preliminary study about POBL, more in-depth theoretical and extensive empirical studies are required in the future. The selection probability of the partial opposite point and its influence also need further justification in the future study.

ACKNOWLEDGMENT

This work was supported in part by the Natural Science Foundation of China under Grant No. 70771042, the Fundamental Research Funds for the Central Universities (2012QN208-HUST), the MOE (Ministry of Education in China) Project of Humanities and Social Science (Project No.13YJA630002), and a grant from the Modern

Information Management Research Center at Huazhong University of Science and Technology. The authors would like to thank Leandro dos Santos Coelho, etc. for making publicly available of the population variance based adaptive Differential Evolution (ADE) implementation.

REFERENCES

[1] H. R. Tizhoosh, “Opposition-Based Learning: A New Scheme for Machine Intelligence,” in *International Conference on Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, 2005, vol. 1, pp. 695–701.

[2] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, “Opposition versus randomness in soft computing techniques,” *Appl. Soft Comput.*, vol. 8, no. 2, pp. 906–918, Mar. 2008.

[3] Mehmet Ergezer and Dan Simon, “Mathematical and Experimental Analyses of Oppositional Algorithms,” *IEEE Trans. Cybern.*, in press.

[4] Q. Xu, L. Wang, N. Wang, X. Hei, and L. Zhao, “A review of opposition-based learning from 2005 to 2012,” *Eng. Appl. Artif. Intell.*, vol. 29, pp. 1–12, Mar. 2014.

[5] F. S. Al-Qunaieer, H. R. Tizhoosh, and S. Rahnamayan, “Opposition based computing– A survey,” in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 2010, pp. 1–7.

[6] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, “A novel population initialization method for accelerating evolutionary algorithms,” *Comput. Math. Appl.*, vol. 53, no. 10, pp. 1605–1614, May 2007.

[7] S. Rahnamayan, G. G. Wang, and M. Ventresca, “An intuitive distance-based explanation of opposition-based sampling,” *Appl. Soft Comput.*, vol. 12, no. 9, pp. 2828–2839, Sep. 2012.

[8] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, “Opposition-Based Differential Evolution,” *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 64–79, 2008.

[9] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, “Opposition-Based Differential Evolution Algorithms,” in *IEEE Congress on Evolutionary Computation, 2006. CEC 2006*, 2006, pp. 2010–2017.

[10] H. Wang, Z. Wu, and S. Rahnamayan, “Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems,” *Soft Comput.*, vol. 15, no. 11, pp. 2127–2140, Nov. 2011.

[11] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, and M. Ventresca, “Enhancing particle swarm optimization using generalized opposition-based learning,” *Inf. Sci.*, vol. 181, no. 20, pp. 4699–4714, Oct. 2011.

[12] H. Wang, “Opposition-Based Barebones Particle Swarm for Constrained Nonlinear Optimization Problems,” *Math. Probl. Eng.*, vol. 2012, Aug. 2012.

[13] M. Ergezer, D. Simon, and D. Du, “Oppositional biogeography-based optimization,” in *IEEE International Conference on Systems, Man and Cybernetics, 2009. SMC 2009*, 2009, pp. 1009–1014.

[14] M. Ergezer and D. Simon, “Oppositional biogeography-based optimization for combinatorial problems,” in *2011 IEEE Congress on Evolutionary Computation (CEC)*, 2011, pp. 1496–1503.

[15] P. K. Roy and D. Mandal, “Quasi-oppositional Biogeography-based Optimization for Multi-objective Optimal Power Flow,” *Electr. Power Compon. Syst.*, vol. 40, no. 2, pp. 236–256, 2011.

[16] P. K. Roy and S. Bhui, “Multi-objective quasi-oppositional teaching learning based optimization for economic emission load dispatch problem,” *Int. J. Electr. Power Energy Syst.*, vol. 53, pp. 937–948, Dec. 2013.

[17] B. Shaw, V. Mukherjee, and S. P. Ghoshal, “A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems,” *Int. J. Electr. Power Energy Syst.*, vol. 35, no. 1, pp. 21–33, Feb. 2012.

[18] S. K. Saha, R. Dutta, R. Choudhury, R. Kar, D. Mandal, and S. P. Ghoshal, “Efficient and Accurate Optimal Linear Phase FIR Filter Design Using Opposition-Based Harmony Search Algorithm,” *Sci. World J.*, vol. 2013, Jun. 2013.

[19] M. El-Abd, “Generalized opposition-based artificial bee colony algorithm,” in *2012 IEEE Congress on Evolutionary Computation (CEC)*, 2012, pp. 1–4.

[20] H.R. Tizhoosh, “Reinforcement Learning Based on Actions and Opposite Actions,” presented at the ICGST International Conference on Artificial Intelligence and Machine Learning (AIML-05), Cairo, Egypt, 2005.

[21] F. Sahba, H. R. Tizhoosh, and M. M. M. A. Salama, “Application of Opposition-Based Reinforcement Learning in Image Segmentation,” in *IEEE Symposium on Computational Intelligence in Image and Signal Processing, 2007. CIISP 2007*, 2007, pp. 246–251.

[22] M. Shokri, H. R. Tizhoosh, and M. Kamel, “Opposition-Based Q(lambda) Algorithm,” in *International Joint Conference on Neural Networks, 2006. IJCNN ’06*, 2006, pp. 254–261.

[23] M. Ventresca and H. R. Tizhoosh, “Improving the Convergence of Backpropagation by Opposite Transfer Functions,” in *International Joint Conference on Neural Networks, 2006. IJCNN ’06*, 2006, pp. 4777–4784.

[24] M. Ventresca and H. R. Tizhoosh, “Numerical condition of feedforward networks with opposite transfer functions,” in *IEEE International Joint Conference on Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence)*, 2008, pp. 3233–3240.

[25] M. Ventresca and H. R. Tizhoosh, “Opposite Transfer Functions and Backpropagation Through Time,” in *IEEE Symposium on Foundations of Computational Intelligence, 2007. FOCI 2007*, 2007, pp. 570–577.

[26] J. J. Liang, B. Y. Qu, P. N. Suganthan, “Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization,” Technical Report 201311, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China And Technical Report, Nanyang Technological University, Singapore, Technical Report, Dec. 2013.

[27] Leandro dos Santos Coelho, Helon V. H. Ayala, Roberto Zanetti Freire, “Population’s Variance-Based Adaptive Differential Evolution for Real Parameter Optimization,” presented at the 2013 IEEE Congress on Evolutionary Computation, Cancún, México, 2013.

[28] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, “Quasi-oppositional Differential Evolution,” in *IEEE Congress on Evolutionary Computation, 2007. CEC 2007*, 2007, pp. 2229–2236.

[29] R. Storn and K. Price, “Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces,” *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[30] J. Vesterstrom and R. Thomsen, “A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems,” in *Congress on Evolutionary Computation, 2004. CEC2004*, 2004, vol. 2, pp. 1980–1987 Vol.2.

[31] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, “Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems,” *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 646–657, 2006.

[32] K. Zielinski, P. Weitkemper, R. Laur, and K.-D. Kammeyer, “Parameter Study for Differential Evolution Using a Power Allocation Problem Including Interference Cancellation,” in *IEEE Congress on Evolutionary Computation, 2006. CEC 2006*, 2006, pp. 1857–1864.