ICIC Express Letters ICIC International ©2016 ISSN 1881-803X
Volume **10**, Number **2**, February **2016** pp. 405–410

SELF-ADAPTIVE DIFFERENTIAL EVOLUTION WITH ELITE OPPOSITION-BASED LEARNING

Zhaolu Guo[1,*], Xuezhi Yue[1], Shenwen Wang[2]
Huogen Yang[1] and Kangshun Li[3]

[1]Institute of Medical Informatics and Engineering
School of Science
Jiangxi University of Science and Technology
No. 86, Hongqi Ave., Ganzhou 341000, P. R. China
*Corresponding author: gzl@whu.edu.cn

[2]School of Information Engineering
Shijiazhuang University of Economics
No. 136, Huaian East Road, Shijiazhuang 050031, P. R. China

[3]School of Mathematics and Informatics
South China Agricultural University
No. 483, Wushan Road, Tianhe District, Guangzhou 510642, P. R. China

Received August 2015; accepted November 2015

Abstract. *Differential evolution (DE) is a very popular stochastic optimization technique, which has achieved many successful applications. In order to improve the efficiency of the traditional DE, this paper presents an enhanced DE, called EOjDE, which utilizes the self-adaptive control parameters scheme and elite opposition-based learning (EOBL) strategy. The self-adaptive control parameters scheme can automatically tune the control parameters according to the characteristics of the problem, while the EOBL strategy is helpful to refine the quality of the individuals in the current population. The proposed EOjDE is evaluated on a set of 13 classical test functions, and is compared with other DE algorithms. The experimental results show the effectiveness and efficiency of the proposed EOjDE.*

Keywords: Evolutionary algorithm, Differential evolution, Self-adaptive, Elite opposition-based learning

1. Introduction.

Differential evolution (DE) is a promising evolutionary algorithm (EA), which has been successfully utilized in various fields [1, 2]. Like other EAs, DE is a population-based stochastic optimization approach. It follows the general framework of an EA, which repeatedly executes the mutation, crossover and selection operators at each generation to create new individuals.

There are two important control parameters in the mutation and crossover operators of DE, namely, scale factor F and crossover rate CR. The scale factor F is related to the search step-size of mutation strategy, while the crossover rate CR controls the number of components that are inherited from the mutation individual. These two control parameters that often greatly influence the performance of DE. However, choosing appropriate values for the control parameters of DE is a time-consuming and difficult task [3]. On the other hand, due to the stochastic nature, DE often suffers from premature convergence and/or slow convergence when tackling complex practical problems [4]. Based on the above considerations, in this paper, we present a self-adaptive differential evolution with elite opposition-based learning strategy, called EOjDE. In order to alleviate the issue of choosing appropriate values for the control parameters, we employ the self-adaptive control parameters schemes [5] to automatically adjust the control parameters according to the feedback from the search process. Moreover, to enhance the exploitation ability, we incorporate the elite opposition-based learning (EOBL) strategy [6] into EOjDE.

The rest of this paper is organized as follows. The basic DE algorithm is introduced in Section 2. The proposed EOjDE algorithm is described in Section 3. Experiments and discussions are presented in Section 4. Finally, Section 5 concludes this paper.

2. **Differential Evolution.**

Like other EAs, DE has a very simple structure. In its search process, DE first initializes a random population with NP individuals, and then it repeatedly performs the mutation, crossover, and selection operators at each generation to steer its population toward the global optimum [7]. The operators of DE are elaborated as follows.

In the mutation step, DE creates a mutation individual $V_i^t = \left[v_{i,1}^t, v_{i,2}^t, \ldots, v_{i,j}^t, \ldots, v_{i,D}^t\right]$ for each individual $X_i^t = \left[x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,j}^t, \ldots, x_{i,D}^t\right]$ (i.e., target individual) in the current population [8], where $i = 1, 2, \ldots, NP$; $j = 1, 2, \ldots, D$; t is the generation, NP is the size of population, and D is the size of decision variables. There are many mutation strategies used in DE algorithms. The most frequently used mutation strategy is DE/rand/1, which is formulated as follows [1]:

$$V_i^t = X_{r1}^t + F \times \left(X_{r2}^t - X_{r3}^t\right) \tag{1}$$

where $r1$, $r2$, and $r3$ are three mutually different indices randomly selected from the set $\{1, 2, \ldots, NP\} \setminus \{i\}$, and $F \in (0, 1)$ is called as scaling factor, scaling the difference vector $X_{r2}^t - X_{r3}^t$.

In the crossover step, DE combines the target individual and the mutation individual with a certain probability to create a trial individual $U_i^t = \left[u_{i,1}^t, u_{i,2}^t, \ldots, u_{i,j}^t, \ldots, u_{i,D}^t\right]$ [1]:

$$u_{i,j}^t = \begin{cases} v_{i,j}^t, & \text{if } \mathrm{rand}(0,1) < CR \text{ or } j == jrand \\ x_{i,j}^t, & \text{otherwise} \end{cases} \tag{2}$$

where rand$(0, 1)$ is a random real number in the range $[0, 1]$, and $jrand$ is a random integer selected from the range $[1, D]$.

After executing the crossover step, DE conducts the selection operator to select the better individual for the next generation between the target individual and the trial individual. For a minimization optimization problem, the selection operator is defined by [2]:

$$X_i^{t+1} = \begin{cases} U_i^t \text{ , if } f(U_i^t) \le f(X_i^t) \\ X_i^t \text{ , otherwise} \end{cases} \tag{3}$$

where $f(.)$ is the minimization objective function.

3. **The Proposed EOjDE.**

3.1. **Operations of EOjDE.**

As pointed out in [9], the two control parameters F and CR often influence the performance of DE. However, there are no constant control parameter values that are suitable for various problems with different characteristics [9]. Moreover, even for a single problem at different evolutionary states, DE still needs different control parameter values to achieve promising performance. Therefore, it is a time-consuming and difficult task to select suitable values for the two control parameters F and CR. To address this issue, various researchers focus on employing adaptive and self-adaptive parameter control strategies in DE [10]. In the adaptive and self-adaptive DE, the control parameters are automatically tuned according to the feedback from the search process. Therefore, the control parameters can dynamically take values that are suitable for the characteristic of the solving problem. In reference [5], a self-adaptive parameter control strategy is proposed to improve the performance of DE. The experimental results show that this self-adaptive parameter control strategy is effective and efficient. Thus, we employ the self-adaptive parameter control strategy in [5] to enhance the performance of our algorithm. In EOjDE, for the ith individual, it has its own control parameters F_i^t and CR_i^t. At each generation, the current control parameters NF_i^t and NCR_i^t for the ith individual are taken values by [5]:

$$NF_i^t = \begin{cases} 0.2 + 0.2 \times \text{rand}(0,1), & \text{if } \text{rand}(0,1) < 0.1 \\ F_i^t, & \text{otherwise} \end{cases} \tag{4}$$

$$NCR_i^t = \begin{cases} 0.8 + 0.2 \times \text{rand}(0,1), & \text{if } \text{rand}(0,1) < 0.1 \\ CR_i^t, & \text{otherwise} \end{cases} \tag{5}$$

After that, the current control parameters NF_i^t and NCR_i^t are used to create mutation and trial individual for the ith individual, respectively. Furthermore, in the selection operator, the control parameters F_i^{t+1} and CR_i^{t+1} associated with the ith individual are updated by:

$$F_i^{t+1} = \begin{cases} NF_i^t, & \text{if } f(U_i^t) < f(X_i^t) \\ F_i^t, & \text{otherwise} \end{cases} \tag{6}$$

$$CR_i^{t+1} = \begin{cases} NCR_i^t, & \text{if } f(U_i^t) < f(X_i^t) \\ CR_i^t, & \text{otherwise} \end{cases} \tag{7}$$

From the above formulas, it is known that the control parameters F and CR of EOjDE can be automatically adjusted according to the feedback from the search process.

Besides the control parameters, the mutation strategy is also very essential for DE. The widely used mutation strategy in DE is DE/rand/1, which can exhibit good exploration ability in the most cases. However, the DE/rand/1 mutation strategy often suffers from poor exploitation, especially at the later stage of evolution. As is known, the population may focus on a small promising region of the fitness landscape at the later stage of evolution. At this stage, the exploitation ability is crucial in finding the optimum solution efficiently [11]. However, the DE/rand/1 strategy often cannot exhibit efficient exploitation ability in this case. To resolve this issue, Wang et al. [6] proposed an elite opposition-based learning (EOBL) strategy to enhance the exploitation of DE. The experimental results indicate that the EOBL strategy has powerful exploitation ability. Therefore, we utilize the EOBL strategy to enhance the exploitation ability of EOjDE. The EOBL strategy is formulated as follows.

For the ith individual X_i^t, its corresponding elite opposition-based individual $EO_i^t = [eo_{i,1}^t, eo_{i,2}^t, \ldots, eo_{i,j}^t, \ldots, eo_{i,D}^t]$ is calculated by [6]:

$$eo_{i,j}^t = K \times \left(EA_j^t + EB_j^t\right) - x_{i,j}^t, \text{ where } K = \text{rand}(0,1) \tag{8}$$

$$EA_j^t = \min\left(ex_{i,j}^t\right), \quad EB_j^t = \max\left(ex_{i,j}^t\right) \tag{9}$$

$$eo_{i,j}^t = \text{rand}(EA_j^t, EB_j^t), \quad \text{if } eo_{i,j}^t < LB_j \text{ or } eo_{i,j}^t > UB_j \tag{10}$$

where $m = 1, 2, \ldots, EN$; $EX_m^t = \left[ex_{m,1}^t, ex_{m,2}^t, \ldots, ex_{m,j}^t, \ldots, ex_{m,D}^t\right]$ are the selected elite individuals used to calculate the search boundaries; $eo_{i,j}^t$ is the elite opposition-based value of $x_{i,j}^t$, EA_j^t and EB_j^t denote the lower and upper boundaries of the jth dimension of the selected elite individuals, respectively; LB_j and UB_j are the lower and upper boundaries of the search space, respectively, and EN represents the size of the selected elite individuals, which is set to $SN * 0.1$, as recommended by the previous work [6].

3.2. **Algorithmic description of EOjDE.** Like GOjDE [5], EOjDE has the similar framework, which combines both the self-adaptive parameter control scheme and elite opposition-based learning strategy into the basic DE. At each generation, EOjDE executes the elite opposition-based learning strategy with probability *eop*, and also performs the self-adaptive DE operations with probability $(1 - eop)$. The framework of EOjDE is described in Algorithm 1, where FEs is the number of fitness evaluations; Max_FEs indicates the maximum number of evaluations, and X_{best} is the global best individual.

Algorithm 1 EOjDE Algorithm

```
1: $t = 0$;
2: $FEs = 0$;
3: Initialize the population;
4: while $FEs < MAX\_FEs$ do
5:     if rand(0,1)< $eop$ then
6:         Select $EN$ elite individuals from the current population $P$;
7:         Calculate the lower and upper boundaries of the selected elite individuals;
8:         Compute the elite opposition-based population $EOP$ according to Equation (8);
9:         Select the fittest $NP$ individuals from $P \bigcup EOP$ to enter the next generation;
10:        $FEs = FEs + NP$;
11:    else
12:        for $i$= 1 to $NP$ do
13:            Randomly select three mutually different indices $r1$, $r2$, $r3$ from the set $\{1, 2, \ldots, NP\} \setminus \{i\}$;
14:            Obtain values for $NF_i^t$ and $NCR_i^t$ according to Equations (4) and (5), respectively;
15:            $jrand$ = randint$(1, D)$;
16:            for $j = 1$ to $D$ do
17:                if rand(0,1)< $NCR_i^t$ or $j == jrand$ then
18:                    $u_{i,j}^t = x_{r1,j}^t + NF_i^t \times (x_{r2,j}^t - x_{r3,j}^t)$;
19:                else
20:                    $u_{i,j}^t = x_{i,j}^t$;
21:                end if
22:            end for
23:            Execute the selection operator according to Equation (3);
24:            Update $F_i^{t+1}$ and $CR_i^{t+1}$ according to Equation (6) and Equation (7), respectively;
25:            $FEs = FEs + 1$;
26:        end for
27:    end if
28:    Save the best individual $X_{best}$;
29:    $t = t + 1$;
30: end while
```

4. Experiments.

4.1. Experimental settings.

In order to testify the effectiveness of the proposed EOjDE, 13 test functions [12] are used to evaluate the efficiency of EOjDE. The size of decision variables D of the test functions is set to 30. In the experiments, EOjDE is compared with DE with self-adapting control parameters (jDE) [4], Opposition-Based DE (ODE) [13], and DE with self-adapting control parameters and generalized opposition-based learning (GOjDE) [5]. For a fair comparison, the common parameters of jDE, ODE, and GOjDE are set to $NP = 100$. The other parameters of jDE, ODE, and GOjDE are set the same as their original papers. In EOjDE, the probability *eop* of the elite opposition-based learning strategy is set to 0.05, following the suggestions in [5]. For each algorithm on each test function, 30 independent runs are conducted with 150,000 function evaluations (FEs) as the stopping criterion. Moreover, the mean and standard deviation of the function error values are recorded for evaluating the efficiency of the algorithms, and two-tailed t-test at a 0.05 significance level [14] is performed on the experimental results.

Table 1. Experimental results of jDE, ODE, GOjDE, and EOjDE over 30 independent runs for the 13 classical test functions

Function	Mean ± SD			
	jDE	ODE	GOjDE	EOjDE
$f1$	1.51E-31±1.82E-31+	6.20E-29±3.92E-29+	4.19E-62±5.91E-62+	2.09E-66±2.80E-66
$f2$	9.13E-19±3.70E-19+	4.31E-09±2.61E-09+	3.28E-33±2.87E-33+	1.33E-34±1.31E-34
$f3$	1.85E-02±6.45E-03+	1.45E-01±1.17E-01+	1.41E-43±1.69E-43+	6.57E-45±9.26E-45
$f4$	3.46E-04±1.23E-04+	1.14E-07±3.43E-07+	2.68E-21±3.72E-21+	7.78E-23±7.78E-23
$f5$	1.87E+01±5.47E-01-	2.29E+01±1.28E+00-	2.79E+01±1.40E-01+	2.77E+01±1.92E-01
$f6$	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
$f7$	5.89E-03±1.45E-03+	1.78E-03±6.21E-04+	1.75E-03±4.34E-04+	1.21E-03±2.16E-04
$f8$	1.34E-02±1.82E-12-	7.51E+03±2.36E+02+	3.90E+03±1.51E+03≈	3.79E+03±1.08E+03
$f9$	0.00E+00±0.00E+00≈	7.83E+01±2.21E+01+	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
$f10$	5.42E-15±1.74E-15+	8.97E-15±1.74E-15+	4.00E-15±0.00E+00≈	4.00E-15±0.00E+00
$f11$	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
$f12$	1.97E-32±8.15E-33-	2.23E-29±2.32E-29-	1.90E-05±2.23E-05+	1.21E-05±1.35E-05
$f13$	2.09E-31±2.93E-31-	2.57E-29±3.07E-29-	7.37E-04±1.03E-03+	3.60E-04±2.41E-04
−	4	3	0	
+	6	8	8	
≈	3	2	5	

Table 2. Average rankings of the four DE algorithms for the 13 test functions achieved by the Friedman test

Algorithm	Ranking
EOjDE	**1.88**
jDE	2.38
GOjDE	2.58
ODE	3.15

4.2. **Results and discussions.** The experimental results are described in Table 1, where the symbols "+", "−", and "≈" denote that EOjDE performs better than, worse than, and similar to the corresponding algorithms according to the two-tailed t-test at a 0.05 significance level, respectively. The results of jDE and ODE are taken from [15]. From Table 1, we can see that EOjDE is significantly better than jDE, ODE, and GOjDE on the majority of the test functions. In particular, EOjDE performs better than jDE, ODE, and GOjDE on 6, 8, and 8 out of 13 test functions, respectively. EOjDE demonstrates similar performance with jDE, ODE, and GOjDE on 3, 2 and 5 test functions, respectively. In addition, jDE and ODE is better than EOjDE only on 4 and 3 test functions, respectively. GOjDE cannot outperform EOjDE on any test function. The excellent performance of EOjDE should be because the self-adaptive control parameters scheme and EOBL strategy can significantly enhance the search ability.

The average ranking of Friedman test is also carried out on the experimental results as recommended by [14]. The average ranking of the four DE algorithms is described in Table 2. The performance of the four DE algorithms can be sorted by the average ranking into the following order: EOjDE, jDE, GOjDE, and ODE. Thus, the best average ranking is achieved by EOjDE, which is better than the other three DE algorithms.

5. **Conclusions.** To promote the search ability of DE, an enhanced DE (EOjDE) is presented in this paper. In EOjDE, it employs the self-adaptive control parameters scheme and elite opposition-based learning (EOBL) strategy to promote the search ability. The control parameters F and CR of EOjDE are automatically tuned by the self-adaptive control parameters scheme, while the exploitation ability of EOjDE is enhanced by the EOBL

strategy. In the experiments, EOjDE is tested on a set of 13 classical test functions, and is compared with three DE variants, namely, jDE, ODE, and GOjDE. The experimental results indicate that EOjDE can achieve better performance on the majority of the test functions.

In the future, we will utilize the proposed EOjDE to solve machine learning problems, such as clustering and regression problems.

Acknowledgment. This work was supported in part by the National Natural Science Foundation of China (Nos. 61402481, 41261093, and 61462036), by Natural Science Foundation of Jiangxi, China (Nos. 20151BAB217010 and 20151BAB201015), and by the Natural Science Youth Foundation of Hebei Educational Committee (No. QN20131053).

REFERENCES

[1] R. Storn and K. Price, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, vol.11, no.4, pp.341-359, 1997.

[2] S. Das and P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, *IEEE Trans. Evolutionary Computation*, vol.15, no.1, pp.4-31, 2011.

[3] A. K. Qin, V. L. Huang and P. N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, *IEEE Trans. Evolutionary Computation*, vol.13, no.2, pp.398-417, 2009.

[4] J. Brest, S. Greiner, B. Bošković, M. Mernik and V. Zumer, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, *IEEE Trans. Evolutionary Computation*, vol.10, no.6, pp.646-657, 2006.

[5] H. Wang, S. Rahnamayan and Z. Wu, Parallel differential evolution with self-adapting control parameters and generalized opposition-based learning for solving high-dimensional optimization problems, *Journal of Parallel and Distributed Computing*, vol.73, no.1, pp.62-73, 2013.

[6] S. Wang, L. Ding, C. Xie, Z. Guo and Y. Hu, A hybrid differential evolution with elite opposition-based learning, *Journal of Wuhan University (Natural Science Edition)*, vol.59, no.2, pp.111-116, 2013.

[7] Y. Wang, Z. Cai and Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, *IEEE Trans. Evolutionary Computation*, vol.15, no.1, pp.55-66, 2011.

[8] Z. Guo, X. Yue, K. Zhang, S. Wang and Z. Wu, A thermodynamical selection-based discrete differential evolution for the 0-1 knapsack problem, *Entropy*, vol.16, no.12, pp.6263-6285, 2014.

[9] J. Zhang and A. C. Sanderson, JADE: Adaptive differential evolution with optional external archive, *IEEE Trans. Evolutionary Computation*, vol.13, no.5, pp.945-958, 2009.

[10] R. Mallipeddi, P. N. Suganthan, Q. K. Pan and M. F. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, *Applied Soft Computing*, vol.11, no.2, pp.1679-1696, 2011.

[11] Z. Guo, Z. Wu, J. Wang, S. Wang and C. Xie, A novel differential evolution algorithm based on elite-cloudy mutation, *Journal of Wuhan University (Natural Science Edition)*, vol.59, no.2, pp.117-122, 2013.

[12] X. Yao, Y. Liu and G. Lin, Evolutionary programming made faster, *IEEE Trans. Evolutionary Computation*, vol.3, no.2, pp.82-102, 1999.

[13] S. Rahnamayan, H. R. Tizhoosh and M. Salama, Opposition-based differential evolution, *IEEE Trans. Evolutionary Computation*, vol.12, no.1, pp.64-79, 2008.

[14] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu and M. Ventresca, Enhancing particle swarm optimization using generalized opposition-based learning, *Information Sciences*, vol.181, no.20, pp.4699-4714, 2011.

[15] Z. Guo, H. Huang, C. Deng, X. Yue and Z. Wu, An enhanced differential evolution with elite chaotic local search, *Computational Intelligence and Neuroscience*, vol.2015, 2015.