An Adaptive Differential Evolution with Learning Parameters According to Groups Defined by the Rank of Objective Values

Tetsuyuki Takahama[1] and Setsuko Sakai[2]

[1] Hiroshima City University, 3-4-1 Ozuka-Higashi, Asaminami-ku, Hiroshima, Japan 731-3194
[2] Hiroshima Shudo University, 1-1-1 Ozuka-Higashi, Asaminami-ku, Hiroshima, Japan 731-3195

Abstract. Differential Evolution (DE) has been successfully applied to various optimization problems. The performance of DE is affected by algorithm parameters such as a scaling factor F and a crossover rate CR. Many studies have been done to control the parameters adaptively. One of the most successful studies on controlling the parameters is JADE. In JADE, the values of each parameter are generated according to one probability density function (PDF) which is learned by the values in success cases where the child is better than the parent. However, search performance might be improved by learning multiple PDFs for each parameter based on some characteristics of search points. In this study, search points are divided into plural groups according to the rank of their objective values and the PDFs are learned by parameter values in success cases for each group.... The advantage of JADE with the group-based learning is shown by solving thirteen benchmark problems.

Keywords: Adaptive differential evolution, Group-based learning, Differential evolution, Evolutionary algorithms

1 Introduction

Optimization problems, especially nonlinear optimization problems, are very important and frequently appear in the real world. There exist many studies on solving optimization problems using evolutionary algorithms (EAs). Differential evolution (DE) is an EA proposed by Storn and Price [9]. DE has been successfully applied to optimization problems including non-linear, non-differentiable, non-convex and multimodal functions [2, 3, 6]. It has been shown that DE is a very fast and robust algorithm.

The performance of DE is affected by algorithm parameters such as a scaling factor F, a crossover rate CR and population size, and by mutation strategies such as a rand strategy and a best strategy. Many studies have been done to control the parameters and the strategies. One of the most successful studies on controlling the parameters is JADE(adaptive DE with optional external archive) [18]. In JADE, the values of parameters F and CR are generated according to

the corresponding probability density function (PDF) and a child is created from the parent using the generated values. The values in success cases, where the child is better than the parent, are used to learn the PDFs. As for F, a location parameter of Cauchy distribution is learned, the scale parameter is fixed and values of F are generated according to the Cauchy distribution. As for CR, a mean of normal distribution is learned, the standard deviation is fixed and values of CR are generated according to the normal distribution. However, search performance might be improved by learning multiple PDFs for F and CR based on some characteristics of search points.

In this study, group-based learning of the PDFs is proposed. Search points are divided into plural groups according to the rank of their objective values. The PDFs are learned by parameter values in success cases for each group. The advantage of JADE with the group-based learning is shown by solving thirteen benchmark problems.

In Section 2, related works are described. DE and JADE are briefly explained in Section 3. In Section 4, JADE with the group-based learning is proposed. The experimental results are shown in Section 5. Finally, conclusions are described in Section 6.

2 Related Works

The performance of DE is affected by control parameters such as the scaling factor F, the crossover rate CR and the population size N, and by mutation strategies such as the rand strategy and the best strategy. Many researchers have been studying on controlling the parameters and the strategies.

The methods of controlling the parameters can be classified into some categories as follows:

(1) selection-based control: Strategies and parameter values are selected regardless of current search state. CoDE(composite DE) [15] generates three trial vectors using three strategies with randomly selected parameter values from parameter candidate sets and the best trial vector will head to the survivor selection.

(2) observation-based control: The current search state is observed, proper parameter values are inferred according to the observation, and parameters and/or strategies are dynamically controlled. FADE(Fuzzy Adaptive DE) [5] observes the movement of search points and the change of function values between successive generations, and controls F and CR. DESFC(DE with Speciation and Fuzzy Clustering) [10] adopts fuzzy clustering, observes partition entropy of search points, and controls CR and the mutation strategies between the rand and the species-best strategy. LMDE(DE with detecting Landscape Modality) [11, 12] detects the landscape modality such as unimodal or multimodal using the change of the objective values at sampling points which are equally spaced along a line. If the landscape is unimodal, greedy parameter settings for local search are selected. Otherwise, parameter settings for global search are selected.

(3) success-based control: It is recognized as a success case when a better search point than the parent is generated. The parameters and/or strategies are adjusted so that the values in the success cases are frequently used. It is thought that the self-adaptation, where parameters are contained in individuals and are evolved by applying evolutionary operators to the parameters, is included in this category. DESAP(DE with Self-Adapting Populations) [14] controls F, CR and N self-adaptively. SaDE(Self-adaptive DE) [7] controls the selection probability of the mutation strategies according to the success rates and controls the mean value of CR for each strategy according to the mean value in success case. jDE(self-adaptive DE algorithm) [1] controls F and CR self-adaptively. JADE(adaptive DE with optional external archive) [18] and MDE_pBX(modified DE with p-best crossover) [4] control the mean or power mean values of F and CR according to the mean values in success cases. CADE(Correlation-based Adaptive DE) [13] introduces the correlation of F and CR to JADE.

In the category (1), useful knowledge to improve the search efficiency is ignored. In the category (2), it is difficult to select proper type of observation which is independent of the optimization problem and its scale. In the category (3), when a new good search point is found near the parent, parameters are adjusted to the direction of convergence. In problems with ridge landscape or multimodal landscape, where good search points exist in small region, parameters are tuned for small success and big success will be missed. Thus, search process would be trapped at a local optimal solution. JADE adopted a weighted mean value for F, which is larger than a usual mean value, and succeeded to reduce the problem of the convergence.

In this study, we propose to improve JADE in the category (3) by introducing group-based learning according to the rank of objective values, which belongs the category (2). Thus, the proposed method is a hybrid method of the category (2) and (3).

3 Optimization by Differential Evolution

3.1 Optimization Problems

In this study, the following optimization problem with lower bound and upper bound constraints will be discussed.

$$\begin{aligned} &\text{minimize} \quad f(\boldsymbol{x}) \\ &\text{subject to } l_j \leq x_j \leq u_j,\ j = 1, \ldots, D, \end{aligned} \tag{1}$$

where $\boldsymbol{x} = (x_1, x_2, \cdots, x_D)$ is a D dimensional vector and $f(\boldsymbol{x})$ is an objective function. The function f is a nonlinear real-valued function. Values l_j and u_j are the lower bound and the upper bound of x_j, respectively.

3.2 Differential Evolution

In DE, initial individuals are randomly generated within given search space and form an initial population of size N. Each individual $\boldsymbol{x}^i, i = 1, 2, \cdots, N$ contains D genes as decision variables. At each generation, all individuals are selected as parents. Each parent is processed as follows: The mutation operation begins by choosing several individuals from the population except for the parent in the processing. The first individual is a base vector. All subsequent individuals are paired to create difference vectors. The difference vectors are scaled by a scaling factor F and added to the base vector. The resulting vector, or a mutant vector, is then recombined with the parent. The probability of recombination at an element is controlled by a crossover rate CR. This crossover operation produces a child, or a trial vector. Finally, for survivor selection, the trial vector is accepted for the next generation if the trial vector is better than the parent.

There are some variants of DE that have been proposed. The variants are classified using the notation DE/*base*/*num*/*cross* such as DE/rand/1/bin and DE/rand/1/exp.

"*base*" specifies a way of selecting an individual that will form the base vector. For example, DE/rand selects an individual for the base vector at random from the population. DE/best selects the best individual in the population.

"*num*" specifies the number of difference vectors used to perturb the base vector. In case of DE/rand/1, for example, for each parent $\boldsymbol{x}^i$, three individuals $\boldsymbol{x}^{p1}$, $\boldsymbol{x}^{p2}$ and $\boldsymbol{x}^{p3}$ are chosen randomly from the population without overlapping $\boldsymbol{x}^i$ and each other. A new vector, or a mutant vector $\boldsymbol{x}'$ is generated by the base vector $\boldsymbol{x}^{p1}$ and the difference vector $\boldsymbol{x}^{p2} - \boldsymbol{x}^{p3}$, where F is the scaling factor.

$$\boldsymbol{x}' = \boldsymbol{x}^{p1} + F(\boldsymbol{x}^{p2} - \boldsymbol{x}^{p3}) \tag{2}$$

"*cross*" specifies the type of crossover that is used to create a child. For example, 'bin' indicates that the crossover is controlled by the binomial crossover using a constant crossover rate, and 'exp' indicates that the crossover is controlled by a kind of two-point crossover using exponentially decreasing the crossover rate.

3.3 JADE

In JADE, the mean value of the scaling factor μ_F and the mean value of the crossover rate μ_{CR} are learned to define two PDFs, where initial values are μ_F=μ_{CR}=0.5. The scaling factor F_i and the crossover rate CR_i for each individual $\boldsymbol{x}^i$ are independently generated according to the two PDFs as follows:

$$F_i \sim C(\mu_F, \sigma_F) \tag{3}$$

$$CR_i \sim N(\mu_{CR}, \sigma_{CR}^2) \tag{4}$$

where F_i is a random variable according to a Cauchy distribution $C(\mu_F, \sigma_F)$ with a location parameter μ_F and a scale parameter σ_F=0.1. CR_i is a random variable according to a normal distribution $N(\mu_{CR}, \sigma_{CR}^2)$ of a mean μ_{CR} and a standard deviation σ_{CR}=0.1. CR_i is truncated to $[0, 1]$ and F_i is truncated to

be 1 if $F_i > 1$ or regenerated if $F_i \leq 0$. The location μ_F and the mean μ_{CR} are updated as follows:

$$\mu_F = (1-c)\mu_F + cS_{F^2}/S_F \tag{5}$$

$$\mu_{CR} = (1-c)\mu_{CR} + cS_{CR}/S_N \tag{6}$$

where S_N is the number of success cases, S_F, S_{F^2} and S_{CR} are the sum of F, F^2 and CR in success cases, respectively. A constant c is a weight of update in (0,1] and the recommended value is 0.1.

JADE adopts a strategy called "current-to-pbest" where an intermediate point between a parent $\boldsymbol{x}^i$ and a randomly selected individual from top individuals is used as a base vector. A mutation vector is generated by current-to-pbest without archive as follows:

$$\boldsymbol{m} = \boldsymbol{x}^i + F_i(\boldsymbol{x}^{pbest} - \boldsymbol{x}^i) + F_i(\boldsymbol{x}^{r2} - \boldsymbol{x}^{r3}) \tag{7}$$

where $\boldsymbol{x}^{pbest}$ is a randomly selected individual from the top $100p\%$ individuals. The child $\boldsymbol{x}^{\text{child}}$ is generated from $\boldsymbol{x}^i$ and $\boldsymbol{m}$ using the binomial crossover.

In order to satisfy bound constraints, a child that is outside of the search space is moved into the inside of the search space. In JADE, each outside element of the child is set to be the middle between the corresponding boundary and the element of the parent as follows:

$$x_j^{\text{child}} = \begin{cases} \frac{1}{2}(l_j + x_j^i) & (x_j^{\text{child}} < l_j) \\ \frac{1}{2}(u_j + x_j^i) & (x_j^{\text{child}} > u_j) \end{cases} \tag{8}$$

This operation is applied when a new point is generated by JADE operations.

4 Proposed method: Group-based learning

In this study, a population of individuals $\{\boldsymbol{x}^i \mid i = 1, 2, \cdots, N\}$ is divided into K groups according to a criterion, where N is the number of individuals and K is the number of groups. All individuals are sorted according to the criterion and the rank r_i ($r_i = 1, 2, \cdots, N$) is assigned to each individual $\boldsymbol{x}^i$. In this study, the objective value of each individual is used as the criterion. The rank of the best individual, who has the best objective value, is 1. In case of $K = 2$, the individuals are divided into good individuals (group 1) and bad individuals (group 2).

The group ID of $\boldsymbol{x}^i$, $group(\boldsymbol{x}^i)$ is defined as follows:

$$group(\boldsymbol{x}^i) = \left\lceil \frac{r_i}{N} K \right\rceil \tag{9}$$

In order to realize group-based learning using parameter control of JADE, the following equations are adopted for each group $k = 1, \cdots, K$.

$$F_i \sim C(\mu_F^k,\, \sigma_F) \tag{10}$$

$$CR_i \sim N(\mu_{CR}^k,\, \sigma_{CR}^2) \tag{11}$$

$$\mu_F^k = (1-c)\mu_F^k + cS_{F^2}^k/S_F^k \tag{12}$$

$$\mu_{CR}^k = (1-c)\mu_{CR}^k + cS_{CR}^k/S_N^k \tag{13}$$

where μ_F^k is the location of Cauchy distribution for F in group k, μ_{CR}^k is the mean of normal distribution for CR in group k. S_N^k is the number of success cases in group k, where the better child than the parent is generated. S_F^k, $S_{F^2}^k$ and S_{CR}^k are the sum of F_i, F_i^2, CR_i at success cases in group k, respectively. As well as JADE, CR_i is truncated to $[0, 1]$ and F_i is truncated to be 1 if $F_i > 1$ or regenerated if $F_i \leq 0$.

5 Numerical Experiments

In this paper, well-known thirteen benchmark problems are solved by the proposed method ADEGL (Adaptive DE with Group-based Learning).

5.1 Test Problems and Experimental Conditions

The 13 scalable benchmark functions are sphere(f_1), Schwefel 2.22(f_2), Schwefel 1.2(f_3), Schwefel 2.21(f_4), Rosenbrock(f_5), step(f_6), noisy quartic(f_7), Schwefel 2.26(f_8), Rastrigin(f_9), Ackley(f_{10}), Griewank(f_{11}), and two penalized functions (f_{12} and f_{13}), respectively [17,18]. Every function has an optimal objective value 0. Some characteristics are briefly summarized as follows: Functions f_1 to f_4 are continuous unimodal functions. The function f_5 is Rosenbrock function which is unimodal for 2- and 3-dimensions but may have multiple minima in high dimension cases [8]. The function f_6 is a discontinuous step function, and f_7 is a noisy quartic function. Functions f_8 to f_{13} are multimodal functions and the number of their local minima increases exponentially with the problem dimension [16].

Experimental conditions are same as JADE as follows: Population size $N = 100$, initial mean for scaling factor $\mu_F = 0.5$ or $\mu_F^k = 0.5$ and initial mean for crossover rate $\mu_{CR} = 0.5$ or $\mu_{CR}^k = 0.5$, the pbest parameter p=0.05, and the learning parameter c=0.1.

Independent 50 runs are performed for 13 problems. The number of dimensions for the problems is 30 (D=30). Each run stops when the number of function evaluations (FEs) exceeds the maximum number of evaluations $FE_{\max}$. In each function, different $FE_{\max}$ is adopted.

5.2 Experimental Results

Table 1 shows the experimental results on JADE, ADEGL (K=2) and ADEGL (K=3). The mean value and the standard deviation of best objective values in 50 runs are shown for each function. The maximum number of function evaluations is selected for each function and is shown in column labeled $FE_{\max}$. The best result among algorithms is highlighted using bold face fonts. Also, Wilcoxon signed rank test is performed and the result for each function is shown under the mean value. Symbols '+', '−' and '=' are shown when ADEGL is significantly better than JADE, is significantly worse than JADE, and is not significantly different from JADE, respectively. Symbols '++' and '−−' are shown when the

significance level is 1% and '+' and '−' are shown when the significance level is 5%.

Table 1. Experimental results on 13 functions

	FE_{max}	JADE	ADEGL (K=2)	ADEGL (K=3)
f_1	150,000	9.38e-59 ± 6.5e-58	**4.32e-66 ± 1.3e-65**	3.36e-64 ± 2.2e-63
			++	++
f_2	200,000	4.19e-31 ± 2.4e-30	5.10e-32 ± 2.7e-31	**2.57e-37 ± 1.6e-36**
			=	++
f_3	500,000	**8.17e-62 ± 3.0e-61**	1.77e-59 ± 1.2e-58	2.25e-60 ± 1.5e-59
			=	=
f_4	500,000	2.01e-23 ± 9.8e-23	**1.20e-24 ± 4.3e-24**	3.70e-24 ± 1.0e-23
			+	=
f_5	300,000	5.78e-01 ± 3.5e+00	**7.97e-02 ± 5.6e-01**	7.26e-01 ± 3.5e+00
			=	=
f_6	10,000	3.02e+00 ± 1.3e+00	**1.78e+00 ± 1.2e+00**	1.98e+00 ± 1.1e+00
			++	++
f_7	300,000	**6.04e-04 ± 2.4e-04**	7.11e-04 ± 2.3e-04	6.80e-04 ± 2.2e-04
			=	=
f_8	100,000	2.37e+00 ± 1.7e+01	**2.46e-05 ± 3.1e-05**	1.18e+01 ± 3.6e+01
			++	+
f_9	100,000	1.01e-04 ± 3.9e-05	**5.64e-05 ± 2.8e-05**	5.95e-05 ± 3.0e-05
			++	++
f_{10}	50,000	9.20e-10 ± 6.4e-10	4.22e-10 ± 3.0e-10	**3.41e-10 ± 3.1e-10**
			++	++
f_{11}	50,000	**1.15e-08 ± 6.9e-08**	1.97e-04 ± 1.4e-03	3.46e-04 ± 1.7e-03
			+	=
f_{12}	50,000	2.40e-16 ± 1.6e-15	4.99e-18 ± 2.6e-17	**1.37e-18 ± 5.5e-18**
			++	++
f_{13}	50,000	1.15e-16 ± 2.2e-16	2.17e-17 ± 5.1e-17	**1.69e-17 ± 7.5e-17**
			++	++
+		—	9	8
=		—	4	5
−		—	0	0

ADEGL (K=2) attained best mean results in 6 functions f_1, f_4, f_5, f_6, f_8 and f_9 out of 13 functions. ADEGL (K=3) attained best mean results in 4 functions f_2, f_{10}, f_{12} and f_{13}. JADE attained best mean results in 3 functions f_3, f_7 and f_{11}. Also, ADEGL (K=2) attained significantly better results than JADE in 9 functions f_1, f_4, f_6, f_8, f_9, f_{10}, f_{11}, f_{12} and f_{13}. ADEGL (K=3) attained significantly better results than JADE in 8 functions f_1, f_2, f_6, f_8, f_9, f_{10}, f_{12} and f_{13}. Thus, it is thought that ADEGL (K=2) is the best method among 3 methods and ADEGL (K=3) is the second best method. JADE could not attain significantly better results than ADEGL of K=2 nor K=3.

Figure 1 shows the change of F and CR over the number of function evaluations for f_1 in case of K=2. ADEGL (K=2) tends to learn smaller values of F and CR than those of JADE for the best group (group 1) and larger values of F and CR than those of JADE for the worst group (group 2).

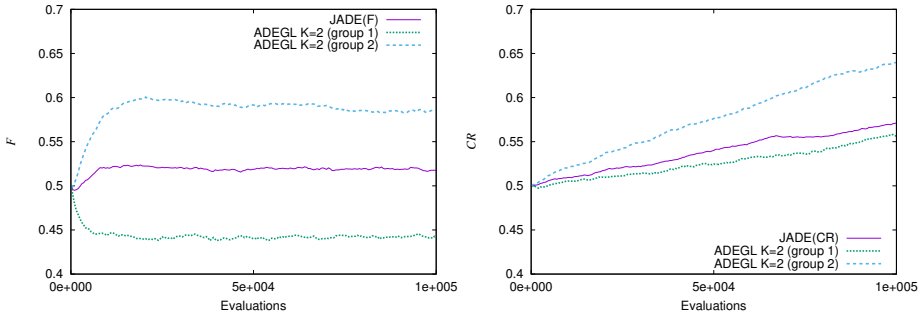

Fig. 1. The graph of F and CR in f_1

6 Conclusion

In this study, group-based learning of algorithm parameters is proposed, where individuals are divided into plural groups according to the rank of objective values and the parameters are learned for each group. DE with group learning is applied optimization of various 13 functions including unimodal functions, a function with ridge structure, multimodal functions. It is shown that the proposed method ADEGL is effective compared with JADE. Also, it is shown that parameters for good individuals are controlled to intensify convergence and parameters for bad individuals are controlled to keep divergence.

In the future, we will apply group-based learning to other adaptive optimization algorithms including differential evolution and particle swarm optimization.

Acknowledgments. This study is supported by JSPS KAKENHI Grant Numbers 26350443 and 17K00311.

References

1. Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Transaction on Evolutionary Computation 10(6), 646–657 (2006)
2. Chakraborty, U.K. (ed.): Advances in Differential Evolution. Springer (2008)
3. Das, S., Suganthan, P.: Differential evolution: A survey of the state-of-the-art. IEEE Transactions on Evolutionary Computation 15(1), 4–31 (2011)

4. Islam, S.M., Das, S., Ghosh, S., Roy, S., Suganthan, P.N.: An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 42(2), 482–500 (2012)
5. Liu, J., Lampinen, J.: A fuzzy adaptive differential evolution algorithm. Soft Computing 9(6), 448–462 (2005)
6. Price, K., Storn, R., Lampinen, J.A.: Differential Evolution: A Practical Approach to Global Optimization. Springer (2005)
7. Qin, A., Huang, V., Suganthan, P.: Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Transactions on Evolutionary Computation 13(2), 398–417 (2009)
8. Shang, Y.W., Qiu, Y.H.: A note on the extended Rosenbrock function. Evolutionary Computation 14(1), 119–126 (2006)
9. Storn, R., Price, K.: Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11, 341–359 (1997)
10. Takahama, T., Sakai, S.: Fuzzy c-means clustering and partition entropy for species-best strategy and search mode selection in nonlinear optimization by differential evolution. In: Proc. of the 2011 IEEE International Conference on Fuzzy Systems. pp. 290–297 (2011)
11. Takahama, T., Sakai, S.: Differential evolution with dynamic strategy and parameter selection by detecting landscape modality. In: Proc. of the 2012 IEEE Congress on Evolutionary Computation. pp. 2114–2121 (2012)
12. Takahama, T., Sakai, S.: Large scale optimization by differential evolution with landscape modality detection and a diversity archive. In: Proc. of the 2012 IEEE Congress on Evolutionary Computation. pp. 2842–2849 (2012)
13. Takahama, T., Sakai, S.: An adaptive differential evolution considering correlation of two algorithm parameters. In: Proc. of the Joint 7th International Conference on Soft Computing and Intelligent Systems and 15th International Symposium on Advanced Intelligent Systems (SCIS&ISIS2014). pp. 618–623 (2014)
14. Teo, J.: Exploring dynamic self-adaptive populations in differential evolution. Soft Computing 10(8), 673–686 (2006)
15. Wang, Y., Cai, Z., Zhang, Q.: Differential evolution with composite trial vector generation strategies and control parameters. IEEE Transactions on Evolutionary Computation 15(1), 55–66 (2011)
16. Yao, X., Liu, Y., , Lin, G.: Evolutionary programming made faster. IEEE Transactions on Evolutionary Computation 3, 82–102 (1999)
17. Yao, X., Liu, Y., Liang, K.H., Lin, G.: Fast evolutionary algorithms. In: Ghosh, A., Tsutsui, S. (eds.) Advances in Evolutionary Computing: Theory and Applications, pp. 45–94. Springer-Verlag New York, Inc., New York, NY, USA (2003)
18. Zhang, J., Sanderson, A.C.: JADE: Adaptive differential evolution with optional external archive. IEEE Transactions on Evolutionary Computation 13(5), 945–958 (2009)