Ensemble of Different Parameter Adaptation Techniques in Differential Evolution

Liang Wang and Wenyin Gong(✉)

School of Computer Science, China University of Geosciences, Wuhan 430074, People's Republic of China
wygong@cug.edu.cn

Abstract. Differential evolution has been proved to be one of the most powerful evolutionary algorithms for the numerical optimization. However, the performance of differential evolution is significantly influenced by its parameter settings. To remedy this limitation, different parameter adaptation techniques are proposed in the literature. Generally, different parameter adaptation techniques have different rationales and may be suitable to different problems. Based on this consideration, in this paper, we attempt to develop the ensemble of different parameter adaptation techniques to enhance the performance of differential evolution. In our proposed method, different parameter adaptation techniques are combined together to adjust the parameters of different solutions in the population. As an illustration, two parameter adaptation techniques proposed in the literature are used in our proposed method. To verify the performance of our proposal, the functions proposed in CEC 2005 are chosen as the test suite. Experimental results indicate that, on the whole, our proposed method is able to provide better results than the single parameter adaptation based differential evolution variants with respect to the non-parametric statistical tests.

1 Introduction

In the field of evolutionary computation, different evolutionary algorithms (EAs) have been developed over the last few decades, such as genetic algorithms, evolution strategies, evolutionary programming, genetic programming [1]. As one of most powerful EAs, differential evolution (DE), proposed by Storn and Price, is very simple and highly efficient for the numerical optimization problems [2]. The main advantages of DE are its simple structure, ease of use, fast convergence speed, etc. Due to these merits, DE has obtained successful applications in diverse fields [3].

In DE, there are three parameters, *i.e.*, population size (μ), crossover rate (Cr), and scaling factor (F), which need to be set properly by the user. However, for a problem at hand, the parameter settings of DE are difficult. More importantly, the performance of DE is significantly influenced by different parameter settings [4]. In EAs, the parameter adaptive control technique is a possible way to remedy the fine-tuning task of parameters [5,6]. Therefore, in the DE literature, different parameter adaptation techniques are developed to enhance its

M. Gong et al. (Eds.): BIC-TA 2016, Part II, CCIS 682, pp. 73–79, 2016.
DOI: 10.1007/978-981-10-3614-9_9

performance, such as jDE [7], SaDE [8], JADE [9], SHADE [10]. Based on the classification of the parameter control techniques in [5], jDE is a self-adaptive control method, while SaDE, JADE, and SHADE belong to the adaptive control methods.

Generally, different parameter control techniques have different features and may be suitable to different problems. In order to solve a wider range of problems, in this paper, the ensemble of different parameter adaptation techniques is proposed, where different parameter control techniques are combined together to adaptively adjust the parameters of different solutions in the population for the problem at hand. As an illustration, two parameter adaptation techniques proposed in jDE [7] and SHADE [10] are used in our proposed framework. The reasons to select the two techniques are two-fold: (i) these two techniques are two different types of parameter control techniques as classified in [5]; and (ii) both of them obtained very promising performance among different DE variants [7,10]. The proposed method is referred to as EADE, *i.e.*, Ensemble of Adaptive DE. To investigate the performance of our proposed EADE, the benchmark functions presented in CEC 2005 are chosen as the test suite [11]. EADE is compared with jDE and SHADE. Experimental results indicate that EADE yields on the whole better results than jDE and SHADE based on the non-parametric statistical test.

2 Our Approach: EADE

In this section, we first present the framework of ensemble of different parameter adaptation techniques. Then, the EADE algorithm is proposed as an illustration, where the parameter adaptation techniques presented in jDE [7] and SHADE [10] are combined together.

2.1 The Framework

Suppose that we have n parameter adaptation techniques (PATs), then, the population $\mathcal{P}$ with μ solutions will be divided into n sub-populations, $\mathcal{P}_1, \cdots, \mathcal{P}_n$, satisfying $\mathcal{P}_1 \cup \cdots \cup \mathcal{P}_n = \mathcal{P}$ and $\mathcal{P}_1 \cap \cdots \cap \mathcal{P}_n = \phi$. Note that the number of solutions in each sub-population may be different, and the mutation strategy can also be different in each sub-population. The framework is plotted in Fig. 1.

Figure 1 indicates that each sub-population has its own PAT to adaptively adjust the parameters of Cr and F for each solution. Although each sub-population has its own PAT, they do not evolve independently: when generating the offspring, the parents (such as $\mathbf{x}_{r1}, \mathbf{x}_{r2}$, and $\mathbf{x}_{r3}$ in "DE/rand/1/bin") in the mutation are chosen from the whole parent population $\mathcal{P}$ as originally used in DE. The main advantage is that it can promote the information sharing for each sub-population. After the offspring population $\mathcal{O}$ is generated, the selection process is performed as originally used in DE. Note that, in the selection process,

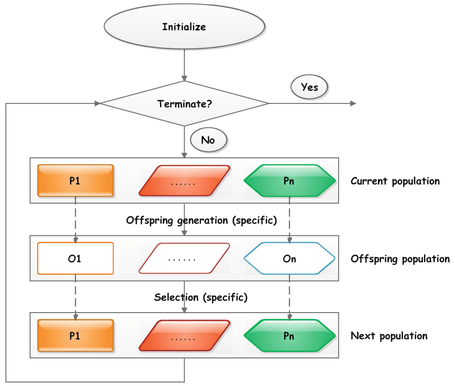

Fig. 1. Framework of ensemble of different parameter adaptation techniques in DE.

the successful parameters need to be saved like JADE [9] and SHADE [10]. In this work, for each PAT the storage of successful parameters is independent, *i.e.*, the successful parameters of each PAT are not influenced mutually.

From above analysis, we can see that the proposed framework still maintains the simple structure of DE. The framework is flexible, different PATs and different mutation strategies can be combined together.

Algorithm 1. The pseudo-code of EADE

Generate the initial population
Evaluate the fitness for each individual
while the halting criterion is not satisfied **do**
  Divide the population into two sub-populations $\mathcal{P}_1$ and $\mathcal{P}_2$
  for $i = 1$ to $|\mathcal{P}_1|$ **do**
    Adaptively update the parameters Cr_i and F_i for each solution using the PAT proposed in jDE [7]
    Generate the trial vector $\mathbf{u}_i$ using "DE/rand/1/bin" strategy as originally used in jDE [7]
  end for
  for $i = |\mathcal{P}_1| + 1$ to μ **do**
    Adaptively update the parameters Cr_i and F_i for each solution using the PAT proposed in SHADE [10]
    Generate the trial vector $\mathbf{u}_i$ using "DE/current-to-$p$best/1/bin" strategy as originally used in SHADE [10]
  end for
  for $i = 1$ to $|\mathcal{P}_1|$ **do**
    Evaluate the offspring $\mathbf{u}_i$
    if $f(\mathbf{u}_i)$ is better than **or** equal to $f(\mathbf{x}_i)$ **then**
      Replace $\mathbf{x}_i$ with $\mathbf{u}_i$
    end if
  end for
  for $i = |\mathcal{P}_1| + 1$ to μ **do**
    Evaluate the offspring $\mathbf{u}_i$
    if $f(\mathbf{u}_i)$ is better than **or** equal to $f(\mathbf{x}_i)$ **then**
      Replace $\mathbf{x}_i$ with $\mathbf{u}_i$
      Store the successful parameters for SHADE
    end if
  end for
  Update the parameters as used in SHADE [10]
end while

2.2 EADE

Based on the above framework, as an illustration, the ensemble adaptive DE (referred to as EADE, in short) is implemented. In EADE, two PATs proposed in jDE [7] and SHADE [10] are used. The reason is that both jDE and SHADE have obtained very promising results in the literature [7,10,12][1]. The pseudo-code of EADE is given in Algorithm 1. In EADE, the parent population $\mathcal{P}$ at each generation is divided into two sub-populations $\mathcal{P}_1$ and $\mathcal{P}_2$. The number of solutions in each sub-population is controlled by the population ratio $\delta \in (0, 1)$, *i.e.*, $|\mathcal{P}_1| = \lfloor \delta \times \mu \rfloor$, and $|\mathcal{P}_2| = \mu - |\mathcal{P}_1|$. In each sub-population, the solutions use their own PAT and mutation strategy. However, the selection of parents to generate the trial vectors are chosen from the whole population to promote the information sharing as mentioned in Sect. 2.1. From Algorithm 1, we can observe that EADE is very similar to jDE and SHADE, however, in EADE both PATs proposed in jDE and SHADE are used. In this way, it can borrow the two merits in both jDE and SHADE as verified in the following experimental results.

3 Experimental Results and Analysis

3.1 Benchmark Functions

To verify the performance of our proposed EADE, 25 benchmark functions presented in CEC 2005 are chosen as the test suite [11]. These functions have been widely used in the literature [10,13,14]. They can be classified into four categories, *i.e.*, unimodal functions (F1–F5), basic multimodal functions (F6–F12), expanded multimodal functions (F13–F14), and hybrid composition multimodal functions (F15–F25). In this work, $D = 30$ is used for all functions.

3.2 Parameter Settings

For all experiments, we use the following parameters for EADE unless a change is mentioned.

- Dimension of each function: $D = 30$;
- Population size: $\mu = 100$;
- Population ratio: $\delta = 0.1$;
- For all functions: $Max\_NFEs = D \times 10,000$.

Note that all other parameters involved in the parameter adaptation techniques in jDE and SHADE are kept the same as originally used in jDE [7] and SHADE [10].

Moreover, in our experiments, each function is optimized over 25 independent runs as suggested in [11]. To avoid any initialization bias, we also use the same set of initial random populations to evaluate different algorithms.

[1] Due to the tight space limitation, jDE and SHADE are not described in this paper. More details can be found in the corresponding references in [7,10], respectively.

3.3 Compared with jDE and SHADE

In this subsection, EADE is compared with jDE and SHADE, because EADE adopts the parameter adaptation techniques proposed in these two algorithms. To make a fair comparison, the parameter settings of jDE and SHADE are set to be the same as used in [7, 10], respectively.

Table 1. Comparison on the errors among EADE, jDE, and SHADE in all functions. All results are averaged over 25 runs. In the last row, the average rankings of different algorithms are obtained by the Friedman test according to the mean values of all functions.

Prob.	EADE		jDE		SHADE	
	Mean	Std	Mean	Std	Mean	Std
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	9.62E-29	9.74E-29	1.26E-05	1.65E-05	**8.46E-29**	8.39E-29
F3	8.46E+03	5.66E+03	1.91E+05	9.38E+04	**7.47E+03**	5.38E+03
F4	**1.07E-15**	2.41E-15	3.97E-01	7.10E-01	1.04E-14	3.70E-14
F5	1.28E-05	3.84E-05	1.03E+03	4.01E+02	**4.38E-07**	1.67E-06
F6	**1.01E-26**	2.02E-26	2.71E+01	2.69E+01	1.59E-01	7.81E-01
F7	**8.37E-03**	6.57E-03	1.31E-02	1.03E-02	1.02E-02	7.70E-03
F8	**2.08E+01**	1.94E-01	2.09E+01	5.36E-02	**2.08E+01**	1.44E-01
F9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	2.17E+01	4.17E+00	5.89E+01	1.02E+01	**2.09E+01**	4.62E+00
F11	**2.61E+01**	1.95E+00	2.80E+01	1.48E+00	2.64E+01	1.67E+00
F12	**1.11E+03**	1.60E+03	1.11E+04	8.55E+03	2.36E+03	3.50E+03
F13	**1.16E+00**	8.18E-02	1.66E+00	1.50E-01	**1.16E+00**	1.00E-01
F14	1.25E+01	2.72E-01	1.30E+01	1.55E-01	**1.24E+01**	2.92E-01
F15	3.28E+02	8.26E+01	**3.27E+02**	1.15E+02	3.44E+02	7.53E+01
F16	**6.81E+01**	7.34E+01	7.60E+01	8.41E+00	8.36E+01	9.82E+01
F17	**9.26E+01**	7.01E+01	1.33E+02	1.36E+01	1.28E+02	1.09E+02
F18	**9.04E+02**	9.97E-01	9.07E+02	1.83E+00	9.05E+02	1.29E+00
F19	**9.04E+02**	1.04E+00	9.07E+02	1.77E+00	9.05E+02	1.17E+00
F20	**9.05E+02**	1.40E+00	9.06E+02	1.89E+00	**9.05E+02**	1.15E+00
F21	5.00E+02	1.14E-13	5.00E+02	1.14E-13	5.00E+02	1.14E-13
F22	**8.63E+02**	1.78E+01	9.08E+02	6.11E+00	8.68E+02	1.80E+01
F23	5.50E+02	7.89E+01	**5.34E+02**	9.80E-05	**5.34E+02**	1.55E-04
F24	2.00E+02	5.91E-13	2.00E+02	0.00E+00	2.00E+02	0.00E+00
F25	**2.09E+02**	1.31E-01	2.10E+02	3.65E-01	**2.09E+02**	1.06E-01
Ranking	**1.56**		2.66		1.78	

The detailed results are reported in Table 1, where the best results are highlighted in **boldface**. In the last row of Table 1, the average rankings of different algorithms are obtained by the Friedman test[2] according to the mean values of all functions. It can be seen that in 14 out of 25 functions EADE is able to get the best mean values. SHADE obtains the best mean values in 10 functions, whereas jDE only provides the best mean values in 2 functions. By carefully looking at the results in Table 1, we observe that SHADE provides better results in unimodal functions, while EADE gets better results in multimodal functions. The reason might be that in EADE the parameter adaptation technique in jDE is combined with "DE/rand/1/bin" strategy, which is less greedy; in this way, EADE enhances the performance in multimodal functions. In addition, according to the averaging rankings by the Friedman test, EADE obtains the best ranking, followed by SHADE and jDE. The *p*-value computed by Iman and Daveport test is $4.88E-05$, which indicates that the results of EADE, jDE, and SHADE are significantly different.

Based on the above results and analysis, we can conclude that ensemble of different parameter adaptation techniques might be useful to enhance the performance of DE variants with single parameter adaptation technique. Ensemble of different parameter adaptation techniques can borrow each of the advantages of different parameter adaptation techniques, and hence, makes the algorithm solve a wider range of problems.

4 Conclusions and Future Work

DE is a simple yet powerful EA when solving the numerical optimization problems. Parameter adaptation is an efficient way to improve the performance of DE. In this paper, we do not propose new parameter adaptation techniques, but present the ensemble of different parameter adaptation techniques. In our approach, different parameter adaptation techniques presented in the literature are cooperated together to evolve the population. As an example, we implement the EADE algorithm, where two parameter adaptation techniques presented in jDE [7] and SHADE [10] are used. Experimental results verified our expectation that EADE improves the performance of jDE and SHADE.

References

1. Bäck, T., Hammel, U., Schwefel, H.P.: Evolutionary computation: comments on the history and current state. IEEE Trans. Evol. Comput. **1**(1), 3–17 (1997)
2. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997)
3. Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. IEEE Trans. Evol. Comput. **15**(1), 4–31 (2011)

[2] Note that, in this work, the non-parametric statistical tests are calculated by the KEEL software [15], which is available online at http://www.keel.es/.

4. Gämperle, R., Müler, S., Koumoutsakos, P.: A parameter study for differential evolution. In: Proceedings of the WSEAS International Conference Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, pp. 293–298 (2002)
5. Eiben, Á.E., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms. IEEE Trans. Evol. Comput. **3**(2), 124–141 (1999)
6. Karafotias, G., Hoogendoorn, M., Eiben, A.E.: Parameter control in evolutionary algorithms: trends and challenges. IEEE Trans. Evol. Comput. **19**(2), 167–187 (2015)
7. Brest, J., Greiner, S., Bošković, B., Mernik, M., Žumer, V.: Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans. Evol. Comput. **10**(6), 646–657 (2006)
8. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans. Evol. Comput. **13**(2), 398–417 (2009)
9. Zhang, J., Sanderson, A.C.: JADE: adaptive differential evolution with optional external archive. IEEE Trans. Evol. Comput. **13**(5), 945–958 (2009)
10. Tanabe, R., Fukunaga, A.: Success-history based parameter adaptation for differential evolution. In: 2013 IEEE Congress on Evolutionary Computation (CEC), pp. 71–78 (2013)
11. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., Tiwari, S.: Problem definitions and evaluation criteria for the CEC_2005 special session on real-parameter optimization (2005)
12. Brest, J., Zumer, V., Maucec, M.: Self-adaptive differential evolution algorithm in constrained real-parameter optimization. In: IEEE Congress on Evolutionary Computation, pp. 215–222 (2006)
13. Wang, Y., Cai, Z., Zhang, Q.: Differential evolution with composite trial vector generation strategies and control parameters. IEEE Trans. Evol. Comput. **15**(1), 55–66 (2011)
14. Gong, W., Cai, Z.: Differential evolution with ranking-based mutation operators. IEEE Trans. Cybern. **43**(6), 2066–2081 (2013)
15. Alcalá-Fdez, J., Sánchez, L., García, S.: KEEL: a software tool to assess evolutionary algorithms to data mining problems. Soft Comput. **13**, 307–318 (2015)