Adaptive Differential Evolution with Optimization State Estimation

Wei-jie Yu and Jun Zhang(Corresponding Author)

Department of Computer Science, Sun Yat-sen University

Key Laboratory of Digital Life, Ministry of Education

Key Laboratory of Software Technology, Education Dept. of Guangdong Province, P.R. China

junzhang@ieee.org

ABSTRACT

The performance of differential evolution (DE) largely depends on an appropriate selection of the values of the algorithmic parameters. Usually, it is difficult to choose optimal parameter values, because they are often ad hoc to the specific problem in question and also related to the optimization states that the DE is in during its search process. In this paper, a novel adaptive parameter control scheme is proposed for DE. Improving from existing parameter control schemes, the parameters *F* and *CR* in DE are adaptively controlled according to the optimization states, namely, exploration state and exploitation state in each generation. These optimization states are estimated by measuring the population distribution. During the optimization process of DE, the distribution of population varies and reflects the search maturity. In the exploration state, individuals in the population distribute evenly in the search space. As the optimization matures, the population gradually converges on a global or local optimum in the exploitation state. This feature enables parameter adaptation with a fuller utilization of the prevailing optimization information and hence reduces inappropriate adjustments. The proposed adaptive parameter control scheme is applied to the famous DE/rand/1 algorithm. Experimental results show that this scheme can effectively improve the efficiency and robustness of the algorithm.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *Heuristic methods*

General Terms

Algorithms

GECCO'12, July 7–11, 2012, Philadelphia, Pennsylvania, USA.

Keywords

Differential evolution, adaptive parameter control, global optimization

1. INTRODUCTION

Differential evolution (DE), first proposed by Storn and Price [1][2], is a simple and efficient evolutionary algorithm (EA) for global optimization. It has been successfully applied to a variety of real-world problems from diverse domains [3][4]. DE involves three general evolutionary operators, i.e. mutation, crossover, and selection, which are associated with certain control parameters. The values of these parameters greatly influence the convergence speed and population diversity . Therefore, how to choose an appropriate parameter setting to improve the performance of the algorithm has become a significant and promising research topic in DE.

A large amount of research work has been conducted to analyze the effects of these control parameters and suggest suitable parameter settings [2][5][6]. However, optimal parameter settings ad hoc to a specific problem are often based on a priori or empirical knowledge, and there exists no single value being good for all types of problems. For example, a small crossover probability *CR* is suitable for separable functions while a large one is effective for non-separable functions [7]. Thus, lots of studies have been undertaken on parameter control for improved DE, where parameters are automatically adjusted at runtime. In the literature, the works on parameter control can be mainly classified into three categories, namely, deterministic, adaptive, and self-adaptive parameter control. The related works will be reviewed in Section 2 of this paper.

However, most existing methods do not explicitly take the runtime state of the optimization into account. For example, parameters are varied mechanistically according to the number of iterations [8]. To improve, different optimization phases should be treated differently. In the early phases of DE optimization, the search direction is undetermined and hence more new regions of the search space should be explored. As the optimization continues, the search direction will be established and the population will gradually converge. In this phase, accelerating convergence and exploitation around some promising points are of the highest importance. Thus, adjusting parameter values according to characteristics of the states of optimization will help enhance the effects of parameter control.

The method of parameter control based on the states of optimization was first proposed by Zhang *et al.* [9], in which a fuzzy system was used to adjust the parameter values for a genetic algorithm (GA). Zhan *et al.* [10] and Yu *et al.* [11] respectively introduced an adaptive particle swarm optimization (APSO) and an adaptive ant colony system (AACS), which also took the evolutionary states into account. However, adaptation strategies for GA, PSO and ACS cannot be applied to DE in the same way.

In this paper, we propose a novel adaptive parameter control scheme for DE based on optimization state estimation. The adaptive parameter control process is implemented mainly in two steps. In the first step, by measuring the population distribution, the optimization state is estimated and thus the search process is classified into one of the two states, i.e., exploration state and exploitation state. The estimation is based on considering the relationship between individuals' fitness values and their distances from the best individual. Then, the values of control parameters F and CR are adaptively adjusted according to the current estimated optimization state. Consequently, the proposed scheme can provide adaptive parameters to match the search requirements of different optimization states.

In order to validate the effect of this adaptive parameter control scheme, we apply it to the famous DE/rand/1 algorithm. By combining DE/rand/1 with parameter adaptation, we develop an adaptive DE algorithm named ADE/rand/1. It is favorably compared with DE/rand/1 on a suite of benchmark functions.

The rest of this paper is organized as follows. Section 2 reviews the DE algorithm and the related works on parameter control methods for DE. Section 3 describes the proposed adaptive parameter control scheme in details, including optimization sate estimation and parameter adjustment. In Section 4, experiments are carried out to verify the effect of the proposed parameter control scheme. Finally, Section 5 draws the conclusions.

2. DE ALGORITHM AND RELATED WORKS

2.1 Differential evolution (DE) algorithm

DE is a population-based stochastic algorithm designed for global numerical optimization. Similar to other EAs, DE searches for a global optimum in the feasible solution space with a population of parameter vectors $\{\mathrm{x}_i^g = [x_{i,1}^g, x_{i,2}^g, ..., x_{i,D}^g], i = 1,2,..., NP\}$, where g denotes the current generation, D is the dimension of the search space, and NP is the population size. In generation g=0, the jth component of the ith vector can be initialized as

$$x_{i,j}^0 = x_{\min,j} + \mathrm{rand}(0,1) \cdot (x_{\max,j} - x_{\min,j}) \quad (1)$$

where rand(0,1) is a uniform random number on the interval [0,1], and $x_{\min,j}$, $x_{\max,j}$ are the prescribed minimum and maximum bounds of the jth dimension, respectively. After initialization, DE enters an evolutionary process which includes mutation, crossover, and selection operations.

Mutation: In each generation g, the mutation operation is applied to each individual x_i^g (also called target vector) to create its corresponding mutant vector v_i^g. The five most frequently used mutation strategies are listed as follows.

- DE/rand/1:

$$\mathrm{v}_i^g = \mathrm{x}_{r1}^g + F \cdot (\mathrm{x}_{r2}^g - \mathrm{x}_{r3}^g) \quad (2)$$

- DE/target-to-best/1:

$$\mathrm{v}_i^g = \mathrm{x}_i^g + F \cdot (\mathrm{x}_{\mathrm{best}}^g - \mathrm{x}_i^g) + F \cdot (\mathrm{x}_{r1}^g - \mathrm{x}_{r2}^g) \quad (3)$$

- DE/best/1:

$$\mathrm{v}_i^g = \mathrm{x}_{\mathrm{best}}^g + F \cdot (\mathrm{x}_{r1}^g - \mathrm{x}_{r2}^g) \quad (4)$$

- DE/best/2:

$$\mathrm{v}_i^g = \mathrm{x}_{\mathrm{best}}^g + F \cdot (\mathrm{x}_{r1}^g - \mathrm{x}_{r2}^g) + F \cdot (\mathrm{x}_{r3}^g - \mathrm{x}_{r4}^g) \quad (5)$$

- DE/rand/2:

$$\mathrm{v}_i^g = \mathrm{x}_{r1}^g + F \cdot (\mathrm{x}_{r2}^g - \mathrm{x}_{r3}^g) + F \cdot (\mathrm{x}_{r4}^g - \mathrm{x}_{r5}^g) \quad (6)$$

It can be seen that the mutant vector v_i^g is generated by combing a base vector with one or two scaled difference vectors. In the above equations, the indices r_1, r_2, r_3, r_4, and r_5 are distinct integers randomly selected from the range [1, NP], and all are different from the index i. $\mathrm{x}_{\mathrm{best}}^g$ is the vector with the best fitness value in the current generation. The factor F is a positive control parameter for amplifying the difference vectors.

Crossover: In order to enhance population diversity, a crossover operation exchanges some components of the mutant vector v_i^g with the target vector x_i^g to generate a trial vector u_i^g. The process can be expressed as

$$u_{i,j}^g = \begin{cases} v_{i,j}^g, & \text{if } \mathrm{rand}(0,1) \le CR \text{ or } j = j_{\mathrm{rand}} \\ x_{i,j}^g, & \text{otherwise} \end{cases} \quad (7)$$

where rand(0,1) is a uniformly distributed random number as before. j_{rand} is an integer randomly generated from the range $[1, D]$, which is used to ensure the trial vector has at least one component different from the target vector. The crossover probability CR is another control parameter, which determines the fraction of vector components inherited from the mutant vector.

Selection: To decide whether the target or the trial vector can survive to the next generation, the selection operation is finally performed. For a minimization problem, the vector with the lower fitness value enters the next generation, which can be expressed as follows:

$$\mathrm{x}_i^{g+1} = \begin{cases} \mathrm{u}_i^g, & \text{if } f(\mathrm{u}_i^g) \le f(\mathrm{x}_i^g) \\ \mathrm{x}_i^g, & \text{otherwise} \end{cases} \quad (8)$$

where $f(\mathrm{x})$ is the objective function for the minimization problem.

2.2 Parameter Control Methods for DE

Control parameters in DE have significant effects on the performance of the algorithm [5][6]. However, there is no fixed parameter setting that can achieve the best performance for all types of problems. Therefore, various parameter control methods have been proposed for DE to dynamically adjust the parameter values. These methods are capable of enhancing the robustness of the DE algorithm. According to the classification by Eiben *et al*. [12], parameter adaptation methods can be classified into three categories as follows.

2.2.1 Deterministic Parameter Control

Deterministic rules change the parameter values without exploiting any information from the evolution. In [8], Das *et al*. proposed two schemes to control the scale factor F of DE. The first one decreases the value of F based on a linear rule, and the second one generates the value of F in a random way. Since the linear rule in the first scheme is based on the current number and the predefined maximum number of generations, it is actually determined before running the algorithm.

2.2.2 Adaptive Parameter Control

By using some form of feedback from the DE search process, adaptive parameter control strategies dynamically adjust the parameter values which can adapt to different evolutionary states. In [13], a fuzzy logic control approach was proposed to adapt the DE parameters F and CR. The fuzzy controllers incorporate the relative fitness values and individuals of the successive generations as their inputs, and the outputs are the values of F and CR. In [14], the value of the parameter F is adaptively adjusted based on the minimum and maximum fitness values over the individuals in each generation. Zaharie [15] proposed a method of adapting the parameters of DE guided by the population diversity evolution. Based on the same idea, Zaharie and Petcu [16] further developed an adaptive Pareto DE for multi-objective optimization problems.

2.2.3 Self-adaptive Parameter Control

Each individual in the population maintains its own set of parameter values, which are encoded into the chromosome and optimized through the evolutionary process. Brest *et al*. [17] introduced a self-adaptive approach for the control parameters F and CR. Each individual in the population is associated with its own parameter values F and CR. In each generation, new values for F and CR are randomly generated in their corresponding ranges with probabilities τ_1 and τ_2, respectively. Qin *et al*. [18] proposed a self-adaptive DE (SaDE) algorithm, in which the trial vector generation strategies as well as the control parameters F and CR are self-adapted by leaning from the previous experiences. Zhang and Sanderson [19] introduced a new adaptive DE called JADE. The control parameters for each individual in JADE are updated based on their historical record of success. More recently, Wang *et al*. [20] proposed a composite DE (CoDE), which uses three trial vector generation strategies and three control parameter settings of F and CR. In each generation, each individual randomly combines these strategies and parameters to generate trial vectors.

3. ADAPTIVE PARAMETER CONTROL SCHEME

In this section, we propose a novel adaptive parameter control scheme for DE. In the parameter adaptation process, the optimization state is first estimated, and then the parameter values F and CR are adjusted accordingly. We can apply this scheme to a classic DE and develop an adaptive DE. Figure 1 shows the process of the adaptive parameter control, and Figure 2 illustrates the flowchart of an adaptive DE algorithm.

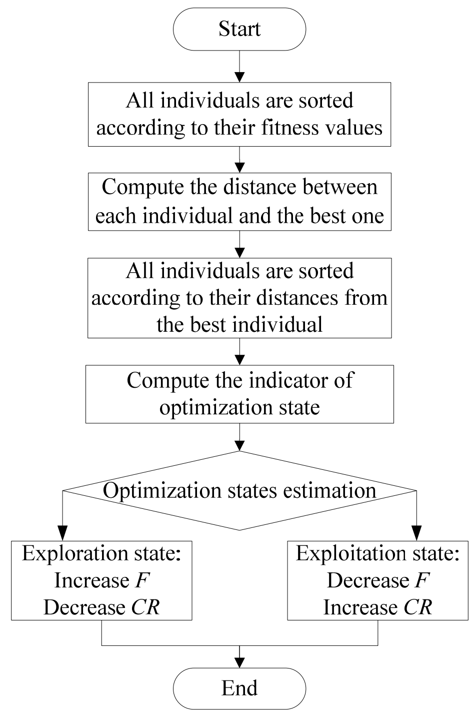

Figure 1. Flowchart of the adaptive parameter control process.

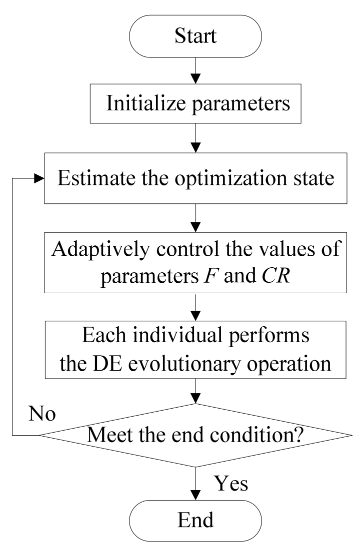

Figure 2. Flowchart of an adaptive DE algorithm.

3.1 Optimization State Estimation

In order to formulate an approach to optimization state estimation for DE, the population distribution characteristics are first described. Generally, the optimization process of DE can be classified into exploration state and exploitation state. In the exploration state, the population distribution is relatively dispersive, since individuals are scattered in the searching space to explore different promising regions. As the optimization progresses, the population will gradually converge, and finally cluster around a global or local optimum in the exploitation state. Due to the variation, the information of population distribution can be used to estimate the optimization state in the DE algorithm. In the following paragraphs, we describe how to measure the population distribution and how to use the distribution information to estimate the optimization state.

In the procedure of optimization state estimation, all individuals are first sorted according to both their fitness values and their distances from the best individual. Further, the relationship between these two sorting orders can be used to measure the population distribution. The detailed steps are as follows.

Step 1): In the beginning of each generation, the fitness values of all the individuals are sorted in a descending order (from the best to the worst). Suppose that the ranking of the fitness value of individual i is denoted as f_i, where $i = 1,2,\ldots,NP$ and NP is the population size.

Step 2): Compute the distances from the best individual to the other individuals. Then, these distances are sorted in an ascending order (from the nearest to the farthest). Suppose that the ranking of the distance of individual i is denoted as d_i, where $i = 1,2,\ldots,NP$.

Step 3): After obtaining the two rankings f_i and d_i for each individual i, compute the indicator of the optimization state (IOS) so as to estimate the current optimization state:

$$IOS = \sum_{i=1}^{NP} |f_i - d_i|, \tag{10}$$

If the two rankings for each individual are exactly the same (i.e., $f_i = d_i$ for each i), indicating that the better individuals are closer to the best individual, IOS has its minimum value:

$$IOS_{\min} = \sum_{r=1}^{NP} |r - r| = 0. \tag{11}$$

On the contrary, if the two rankings for each individual are just the opposite (i.e., $f_i+d_i = NP+1$ for each i), indicating that the better individuals are farther from the best individual, IOS has its maximum value:

$$IOS_{\max} = \begin{cases} \dfrac{NP \cdot NP}{2}, & \text{if } NP \text{ is even} \\ \dfrac{(NP+1)\cdot(NP-1)}{2}, & \text{if } NP \text{ is odd} \end{cases}. \tag{12}$$

Step 4): The value of IOS is normalized by the difference between the values of $IOS_{\max}$ and $IOS_{\min}$ as

$$IOS' = \frac{IOS - IOS_{\min}}{IOS_{\max} - IOS_{\min}}, \tag{13}$$

where IOS' is the normalized value of IOS, ranging from zero to one.

Step 5): Perform the estimation of the optimization state. According to the value of IOS', the optimization process is classified into one of the two optimization states, i.e., exploration state and exploitation state:

$$\Phi = \begin{cases} S_1, & \text{if } \operatorname{rand}(0,1) < IOS' \\ S_2, & \text{otherwise} \end{cases}, \tag{14}$$

where Φ is the estimated optimization state, S_1 and S_2 represent the exploration state and exploitation state, respectively, and rand(0,1) is a uniform random number within [0,1]. According to (14), the optimization process has a probability of IOS' and (1– IOS') to be classified into exploration state and exploitation state, respectively. If the value of IOS' is large, the differences between the two rankings f_i and d_i are obvious. In this case, there exist many good individuals far away from the best individual, which means that the population is exploring different promising regions. Thus, the optimization process is more likely to be in the exploration state. On the contrary, if the value of IOS' is small, most of the good individuals have converged around the best individual, and thus the optimization process has a large probability to be in the exploitation state.

3.2 Parameter Adjustment

Before describing the detail of the control scheme, we first briefly discuss the effects of the parameters F and CR. According to (9), the control parameter F is used to scale the difference vector. Using a large value of F generates a mutant vector largely different from the base vector chosen from the population, and thus helps maintain the population diversity. In contrast, a small value of F is more likely to facilitate convergence. According to (7), CR is the probability that a vector component will be inherited from the mutant vector. Therefore, a large value of CR can speed up convergence [2] [5], whereas a small value of CR is good for preventing premature convergence of the population.

Based on the above considerations, the strategies for adjusting the control parameter values F and CR in different optimization states are defined as follows.

Exploration State – Increasing F and Decreasing CR: In the exploration state, in order to explore more promising regions, it is better to increase the value of F. Conversely, the value of CR should be decreased so that premature convergence of the population can be avoided.

Exploitation State – Decreasing F and Increasing CR: In the exploitation state, in order to accelerate the convergence, decreasing of the value of F is an appropriate way. Meanwhile, increasing the value of CR can help increase the convergence speed.

Based on the above strategies, the values of F and CR can be adjusted adaptively according to the current estimated optimization state Φ. The adjustment is based on the values of F and CR of the previous generation, as shown in (15) and (16)

$$F(g) = \begin{cases} F(g-1) + c_F \cdot \Delta F, & \text{if } \Phi = S_1 \\ F(g-1) - c_F \cdot \Delta F, & \text{if } \Phi = S_2 \end{cases}, \quad (15)$$

$$CR(g) = \begin{cases} CR(g-1) - c_{CR} \cdot \Delta CR, & \text{if } \Phi = S_1 \\ CR(g-1) + c_{CR} \cdot \Delta CR, & \text{if } \Phi = S_2 \end{cases}, \quad (16)$$

where

$$\Delta F, \Delta CR = \begin{cases} \dfrac{IOS - IOS_{\min}}{IOS_{\max} - IOS_{\min}}, \text{if } \Phi = S_1 \\ \dfrac{IOS_{\max} - IOS}{IOS_{\max} - IOS_{\min}}, \text{if } \Phi = S_2 \end{cases}. \quad (17)$$

The coefficients c_F and c_{CR} are set as 0.1. According to (17), the step of adjustment is related to the value of *IOS*. The values of *F* and *CR* are both clamped in the range of [0,1].

4. EXPERIMENTAL STUDIES

4.1 Benchmark Functions and Experimental Setup

In this section, experiments are carried out to evaluate the effect of the proposed adaptive parameter control scheme. We use 8 benchmark functions which are listed in Table 1 [21]. Functions f_1 is a unimodal function. This function can be used to test the convergence speed of the algorithms. Functions f_2 is the Rosenbrock function which is unimodal for $D \leq 3$, but may become multimodal when the dimension is high [22]. Functions f_3-f_8 are multimodal functions where the number of local optima increases exponentially with the problem dimension. Such functions appear to be the most difficult class of problems for many optimization algorithms. The algorithms' global search ability to escape from local optima can be verified by these multimodal functions.

We apply the adaptive parameter control scheme to DE with DE/rand/1 mutation strategy. The new adaptive DE algorithm is denoted as ADE/rand/1. Three DE/rand/1 algorithms with different parameter settings are compared with ADE/rand/1. All the classic DEs set *F* to 0.5 as suggested in most of the literatures [2][6], and set *CR* to 0.1, 0.5, and 0.9 respectively.

For a fair comparison, all the compared DE algorithms use a population size of 100. Moreover, each algorithm is run 25 times independently and the results are averaged. For clarity, the results of the best algorithms are marked in **boldface**.

Table 1. Benchmark functions used in this paper

Name	Test function	D	S	f_{min}
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	30	$[-100, 100]^D$	0
Rosenbrock	$f_2(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]^D$	0
Schwefel	$f_3(x) = \sum_{i=1}^{D} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30	$[-500, 500]^D$	-12569.5
Rastrigin	$f_4(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]^D$	0
Ackley	$f_5(x) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i) + 20 + e$	30	$[-32, 32]^D$	0
Griewank	$f_6(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$[-600, 600]^D$	0
Penalized	$f_7(x) = \frac{\pi}{D}\{10\sin^2(\pi y_i) + \sum_{i-1}^{D-1}(y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1})] + (y_D - 1)^2\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4),$	30	$[-50, 50]^D$	0
Penalized	$f_8(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2[1 + \sin 2(3\pi x_{i+1})] + (x_D - 1)^2[1 + \sin^2(2\pi x_D)]\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$	30	$[-50, 50]^D$	0

4.2 Effect of the Adaptive Parameter Control Scheme

The experimental results of classic DEs and adaptive DEs are listed in Table 2. It can be seen that ADE/rand/1 generally outperforms DE/rand/1. Moreover, the performance of DE/rand/1 is very sensitive to the parameter settings. For example, the DE/rand/1 (CR = 0.1) is able to find the near-global optimum on most of the multimodal functions, but it performs the worst on the unimodal function f_2. In contrast, the DE/rand/1 (CR = 0.5) obtains the highest accuracy of results on the unimodal function f_2, but it suffers from frequent premature convergence on the multimodal function f_4. The performance of ADE/rand/1 is less dependent on the optimization problems. Not only can it get high accuracy solutions on unimodal functions, but also it has strong global search ability to escape from local optima on multimodal functions. These results demonstrate that our adaptive parameter control scheme is helpful to improve robustness of the algorithm. This is because the parameter adaptation scheme is able to adapt the control parameters to match different characteristics of different problems.

Table 2. Comparison between ADE/rand/1 and DE/rand/1

Fun.	FEs.	ADE/rand/1 Mean (Std Dev)	DE/rand/1 (CR=0.1) Mean (Std Dev)	DE/rand/1 (CR=0.5) Mean (Std Dev)	DE/rand/1 (CR=0.9) Mean (Std Dev)
f_1	150000	**4.92E-28** **(1.84E-27)**	3.25E-19 (1.11E-19)	1.07E-17 (4.39E-18)	2.03E-16 (1.85E-16)
f_2	2000000	9.36E-01 (8.71E-01)	2.25E+01 (2.07E+00)	**3.44E-17** **(1.23E-16)**	1.59E-01 (7.81E-01)
f_3	900000	**-12569.49** **(1.82E-12)**	**-12569.49** **(1.82E-12)**	**-12569.49** **(1.82E-12)**	-12093.07 (7.33E+02)
f_4	500000	**0.00E+00** **(0.00E+00)**	**0.00E+00** **(0.00E+00)**	9.62E+01 (7.21E+00)	7.27E+01 (2.34E+01)
f_5	200000	**4.71E-15** **(1.30E-15)**	2.90E-14 (3.48E-15)	2.01E-13 (4.80E-14)	2.18E-12 (1.18E-12)
f_6	200000	**0.00E+00** **(0.00E+00)**	**0.00E+00** **(0.00E+00)**	**0.00E+00** **(0.00E+00)**	**0.00E+00** **(0.00E+00)**
f_7	150000	**3.03E-26** **(1.13E-25)**	8.50E-21 (2.66E-21)	1.94E-17 (1.24E-17)	5.02E-17 (6.65E-17)
f_8	150000	**1.30E-24** **(3.91E-24)**	5.41E-20 (2.00E-20)	6.01E-17 (5.28E-17)	1.85E-16 (1.92E-16)

5. CONCLUSION

In this paper, a novel adaptive parameter control scheme for DE has been proposed. The control parameters F and CR in DE are adaptively controlled based on optimization state estimation. We have applied the proposed scheme to the famous DE/rand/1 and thus developed an adaptive DE named ADE/rand/1. Experimental results showed that the adaptive parameter control scheme can effectively improve the performance of the algorithms on both unimodal and multimodal problems. For future work, we will apply the adaptive parameter control scheme to other evolutionary algorithms [23], such as particle swarm optimization (PSO) [24] and ant colony optimization (ACO) [25].

6. ACKNOWLEDGMENTS

This work was supported in part by the National Science Fund for Distinguished Young Scholars No.61125205, National Natural Science Foundation of China No.61070004, and NSFC Joint Fund with Guangdong under Key Project U0835002.

7. REFERENCES

[1] R. M. Storn and K. V. Price, “Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces,” TR-95-012, 1995 [Online]. Available: http://icsi.berkeley.edu/~storn/litera.html.

[2] R. M. Storn and K. V. Price, “Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces,” *J. Global Optim.*, vol. 11, pp. 341–359, 1997.

[3] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Berlin, Germany: Springer-Verlag, 2005.

[4] S. Das and P. N. Suganthan, “Differential evolution: a survey of the state-of-the-art,” *IEEE Trans Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[5] R. Gämperle, S. D. Müller, and P. Koumoutsakos, “A parameter study for differential evolution,” in *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, A. Grmela and N. E. Mastorakis, Eds. Interlaken, Switzerland: WSEAS Press, 2002, pp. 293–298.

[6] J. Rönkkönen, S. Kukkonen, and K. V. Price, “Real-parameter optimization with differential evolution,” in *Proc. IEEE Congr. Evol. Comput.*, 2005, pp. 506–513.

[7] A. M. Sutton, M. Lunacek, and L. D. Whitley, “Differential evolution and non-separability: Using selective pressure to focus search,” in *Proc. Genetic Evol. Comput. Conf.*, Jul. 2007, pp. 1428–1435.

[8] S. Das, A. Konar, and U. K. Chakraborty, “Two improved differential evolution schemes for faster global search,” in *Proc. Genetic Evol. Comput. Conf.*, 2005, pp. 991–998.

[9] J. Zhang, H. Chung, and W. L. Lo, “Clustering-based adaptive crossover and mutation probabilities for genetic algorithms,” *IEEE Trans. Evolutionary Computation*, vol. 11, no. 3, pp. 326-335, June 2007.

[10] Z.-H. Zhan, J. Zhang, Y. Li, and H. Chung, “Adaptive particle swarm optimization,” *IEEE Trans. Systems, Man, and Cybernetics-Part B*, vol. 39, no. 6, pp. 1362-1381, Dec. 2009.

[11] W.-J Yu and J. Zhang, “Pheromone-distribution-based adaptive ant colony system,” in *Proc. Genetic Evol. Comput. Conf.*, Portland, America, 2010, pp. 31-38.

[12] A. E. Eiben, R. Hinterding, and Z. Michalewicz, “Parameter control in evolutionary algorithms,” *IEEE Trans. Evolutionary Computation*, vol. 3, no. 2, pp. 124-141, July 1999.

[13] J. Liu and J. Lampinen, “A fuzzy adaptive differential evolution algorithm,” *Soft Comput. – A Fusion of Foundation, Methodologies and Applications*, vol. 9, no. 6, pp. 448–462, Apr. 2005.

[14] M. M. Ali and A. Törn, "Population set based global optimization algorithms: Some modifications and numerical studies," *Comput. Oper. Res.*, vol. 31, no. 10, pp. 1703–1725, 2004.

[15] D. Zaharie, "Control of population diversity and adaptation in differential evolution algorithms," In D. Matousek, P. Ošmera (eds.), *Proc. of MENDEL* 2003, *In 9th International Conference on Soft Computing, Brno*, Czech Republic, pp. 41–46, June 2003.

[16] D. Zaharie and D. Petcu, "Adaptive Pareto differential evolution and its parallelization," *Proc. of 5th International Conference on Parallel Processing and Applied Mathematics*, Czestochowa, Poland, Sept. 2003.

[17] 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, Dec. 2006.

[18] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.

[19] J. Q. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.

[20] Y. Wang, Z. X. Cai, and Q. F. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55–66, Feb. 2011.

[21] X. Yao, Y. Liu, and G. M. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

[22] Y.-W. Shang and Y.-H. Qiu, "A note on the extended rosenbrock function," *Evol. Comput.*, vol. 14, no. 1, pp. 119–126, 2006.

[23] J. Zhang, Z.-H. Zhan, Y. Lin, N. Chen, Y.-J. Gong, H. Chung, Y. Li and Y.-H. Shi, "Evolutionary Computation Meets Machine Learning: A Survey", *IEEE Computational Intelligence Magazine*, pp. 68–75, Nov. 2011.

[24] W. -N. Chen, J. Zhang, H. Chung, W. -L. Zhong, W. -G. Wu and Y. -H. Shi, "A novel set-based particle swarm optimization method for discrete optimization problems", *IEEE Trans. Evol. Comput.* vol. 14, no. 2, pp. 278–300, Apr. 2010.

[25] W. -N. Chen and J. Zhang, "Ant colony optimization approach to grid workflow scheduling problem with various QoS requirements", *IEEE Trans. Systems, Man, and Cybernetics-Part C*, vol. 31, no. 1, pp. 29–43, Jan. 2009.